

**UniSenac**  
Centro Universitário RS



# Sistemas Operacionais

# LOGIC CIRCUIT

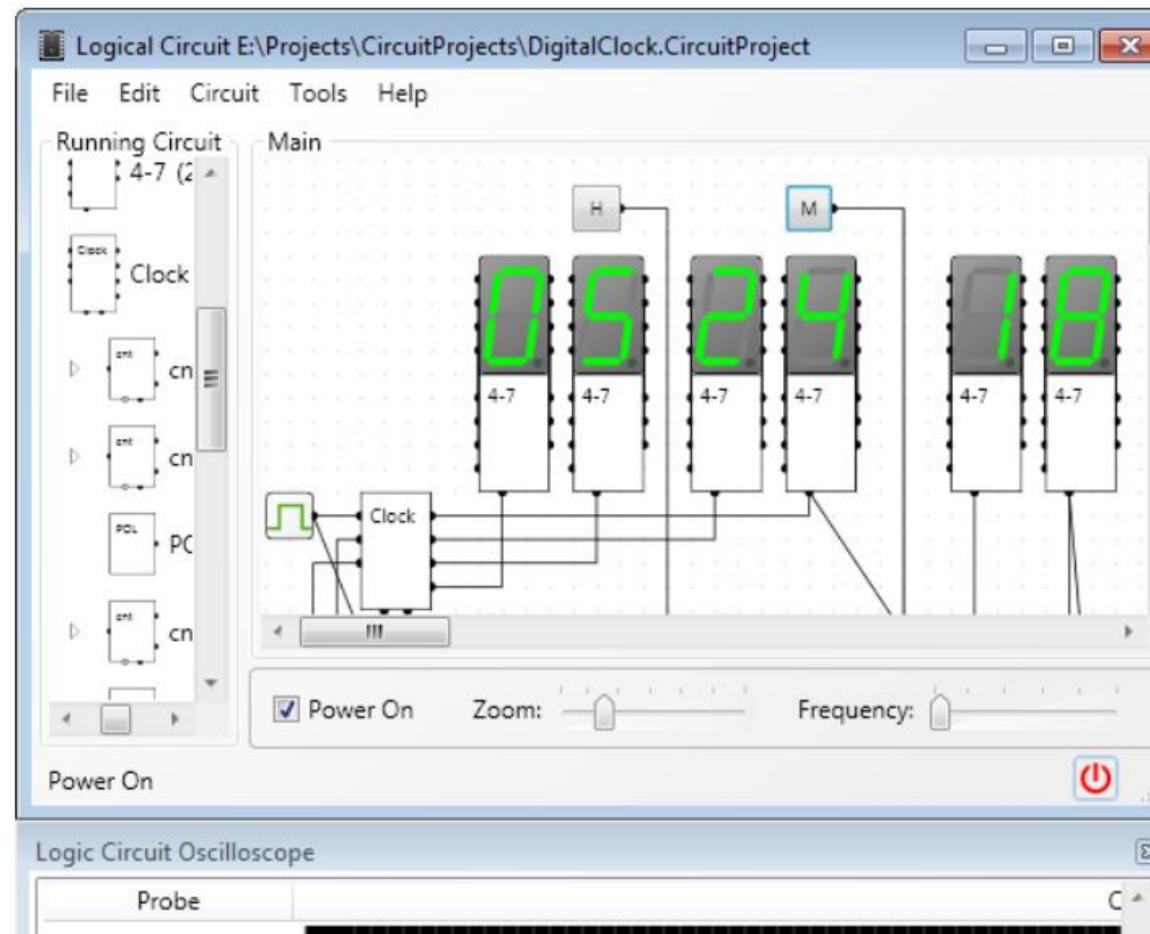
LogicCircuit - is educational software for designing and simulating digital logic circuits.



Home   Download   Screenshots   Contact   Help   Forum

## Logic Circuit

**LogicCircuit** – is free, open source educational software for designing and simulating digital logic circuits. Intuitive graphical user interface, allows you to create unrestricted circuit hierarchy with multi bit buses, debug circuits behavior with oscilloscope, and navigate running circuits hierarchy.



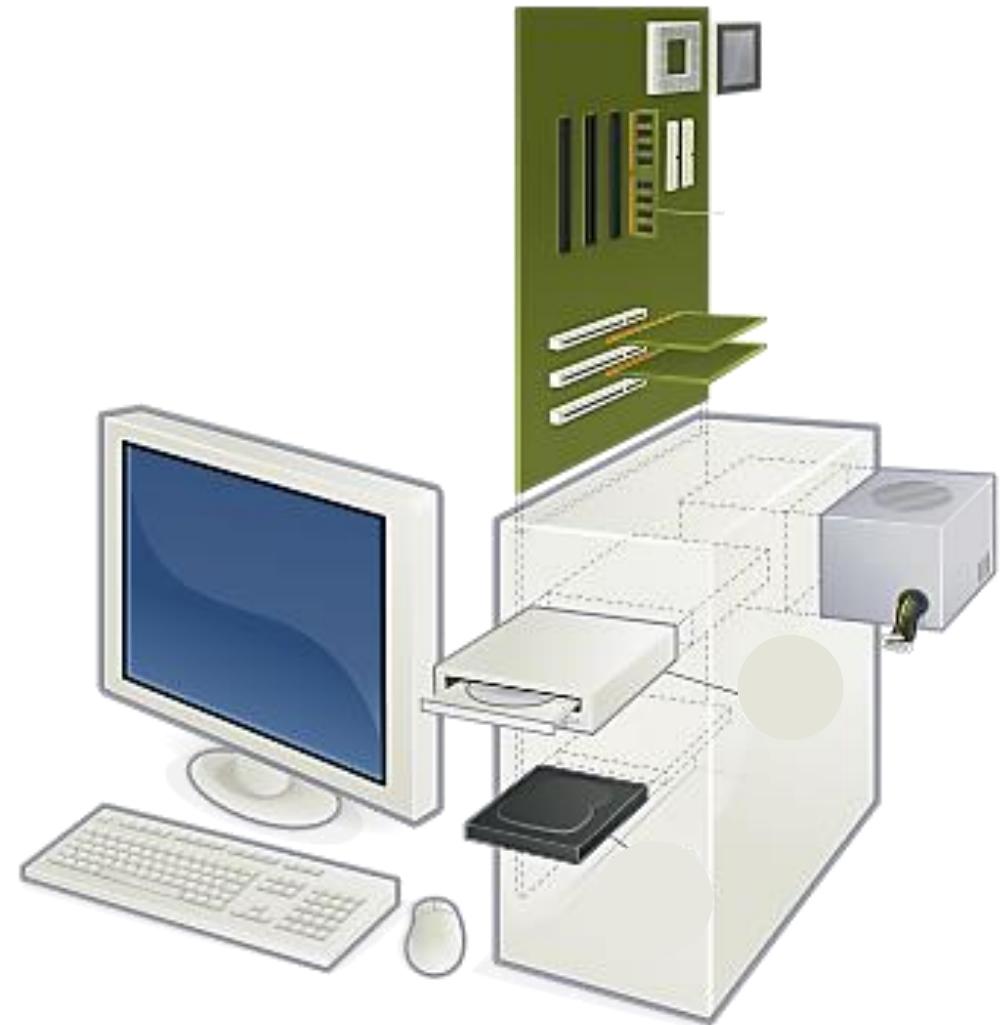
<https://www.logiccircuit.org>

<http://177.101.203.139/gladimir/tools>

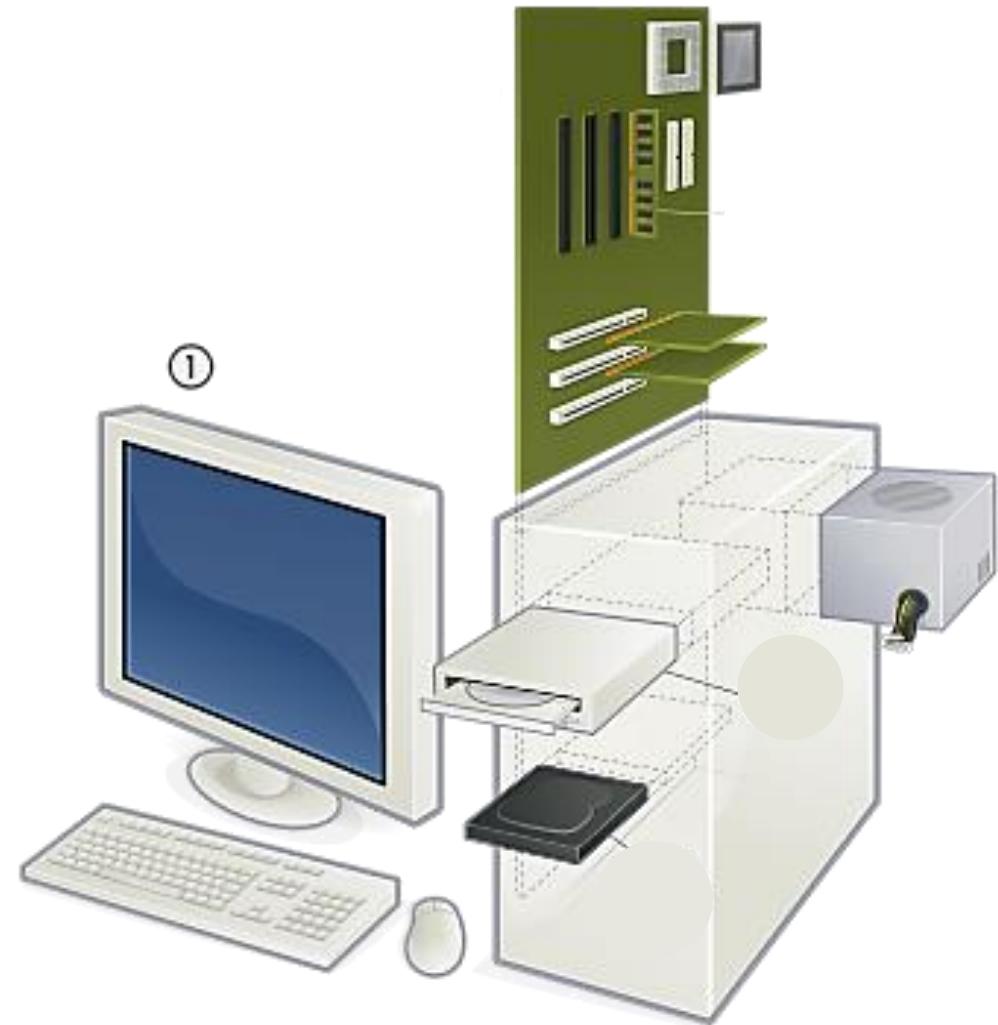
<http://192.168.200.3/gladimir/tools>

# Hardware - Review

# Hardware - Review

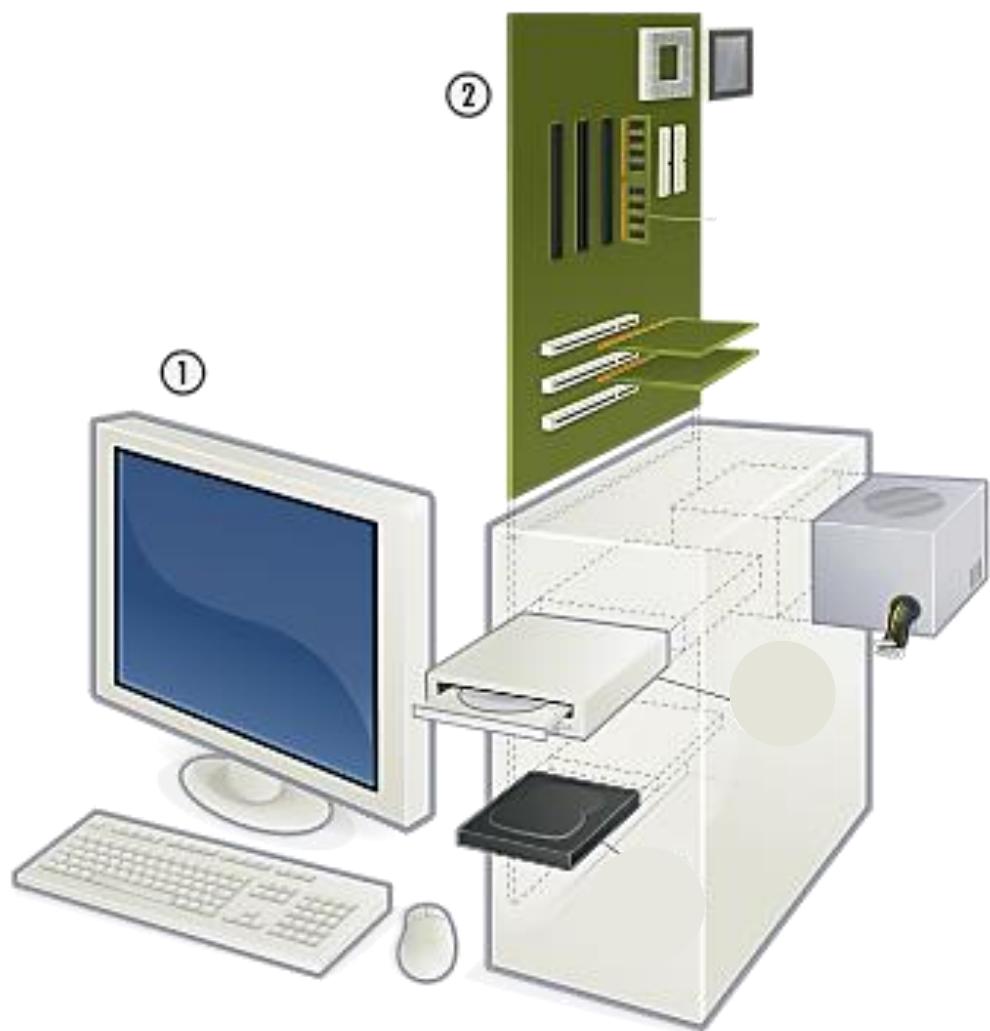


# Hardware - Review



# Hardware - Review

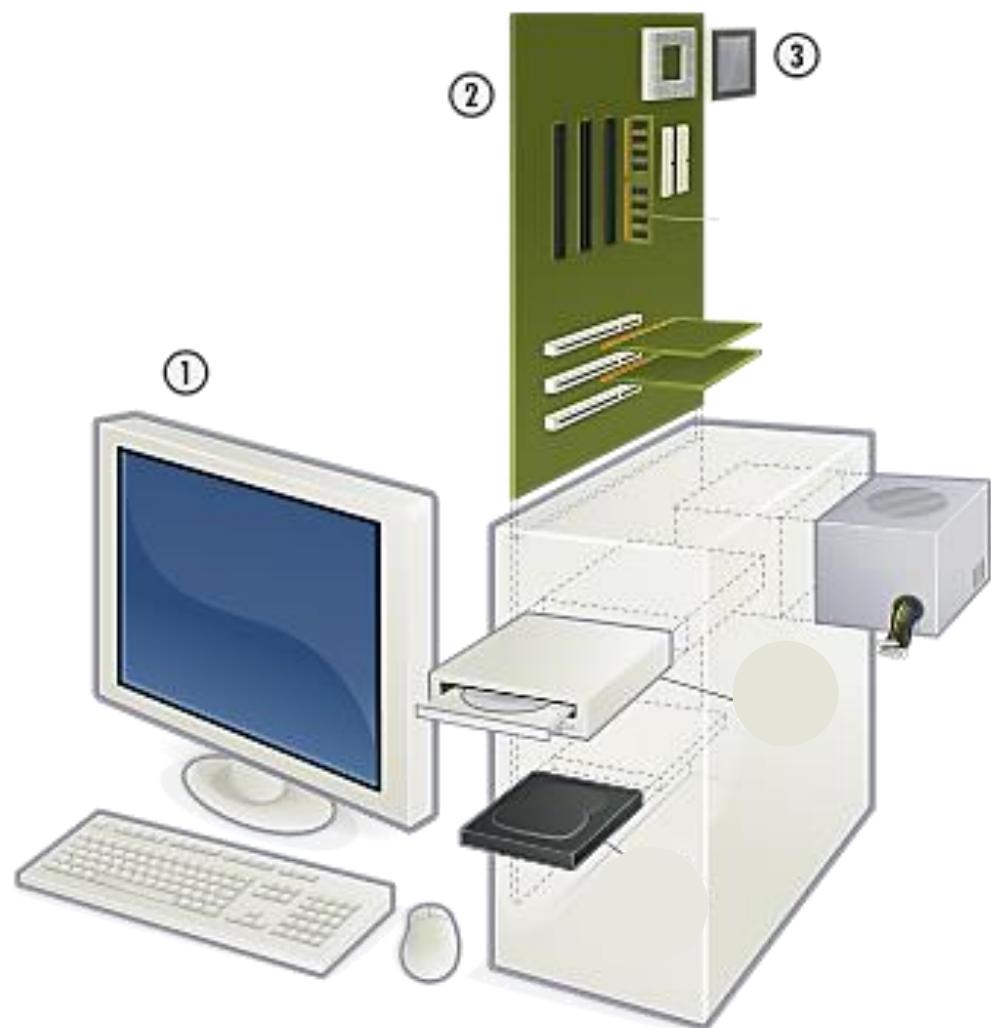
## 01 - Monitor



# Hardware - Review

01 - Monitor

02 - Placa-Mãe

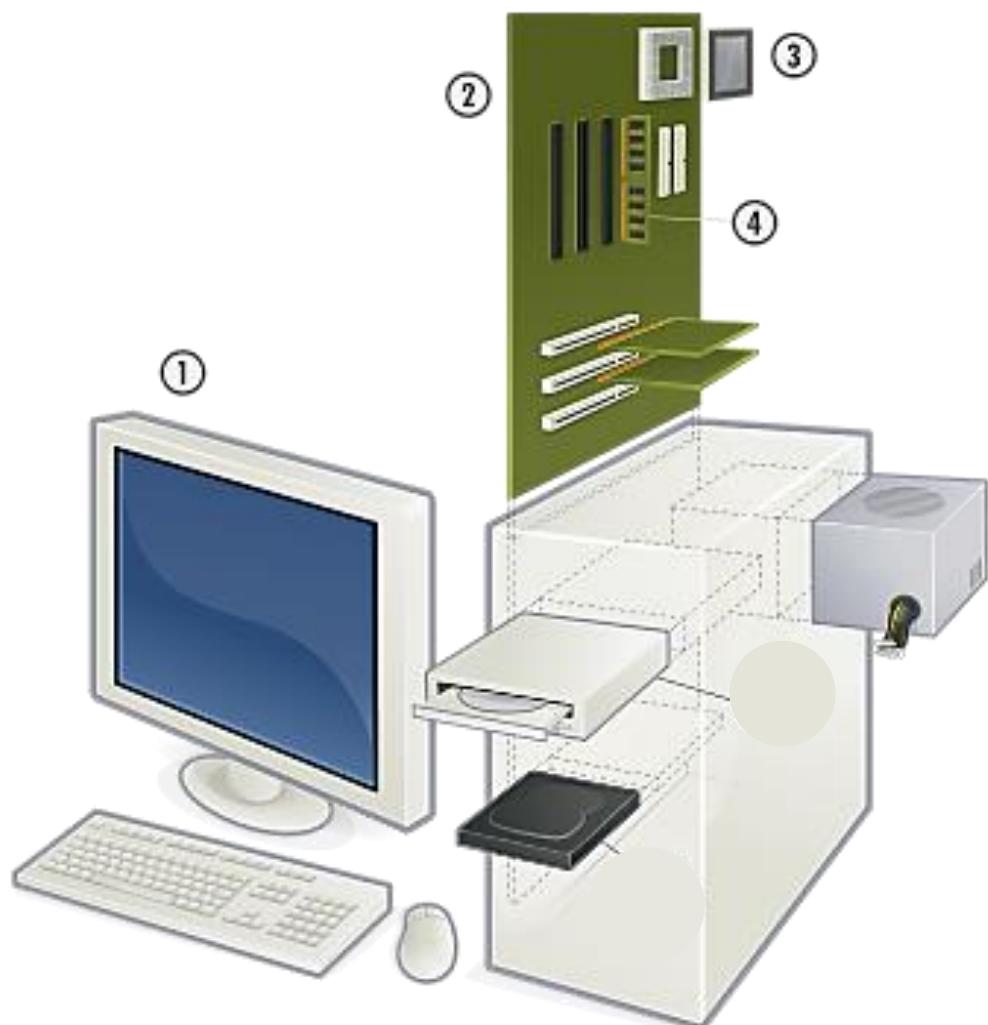


# Hardware - Review

01 - Monitor

02 - Placa-Mãe

03 - Processador



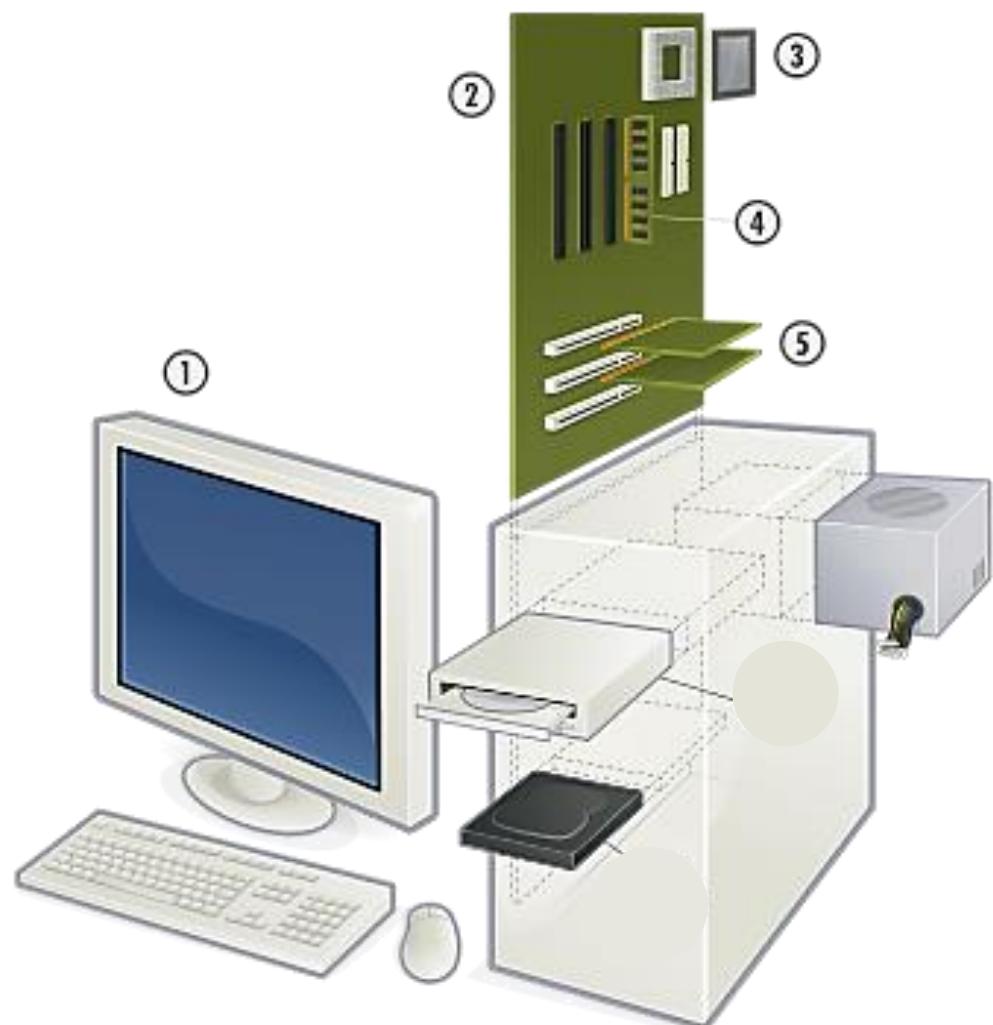
# Hardware - Review

01 - Monitor

02 - Placa-Mãe

03 - Processador

04 - Memória RAM



# Hardware - Review

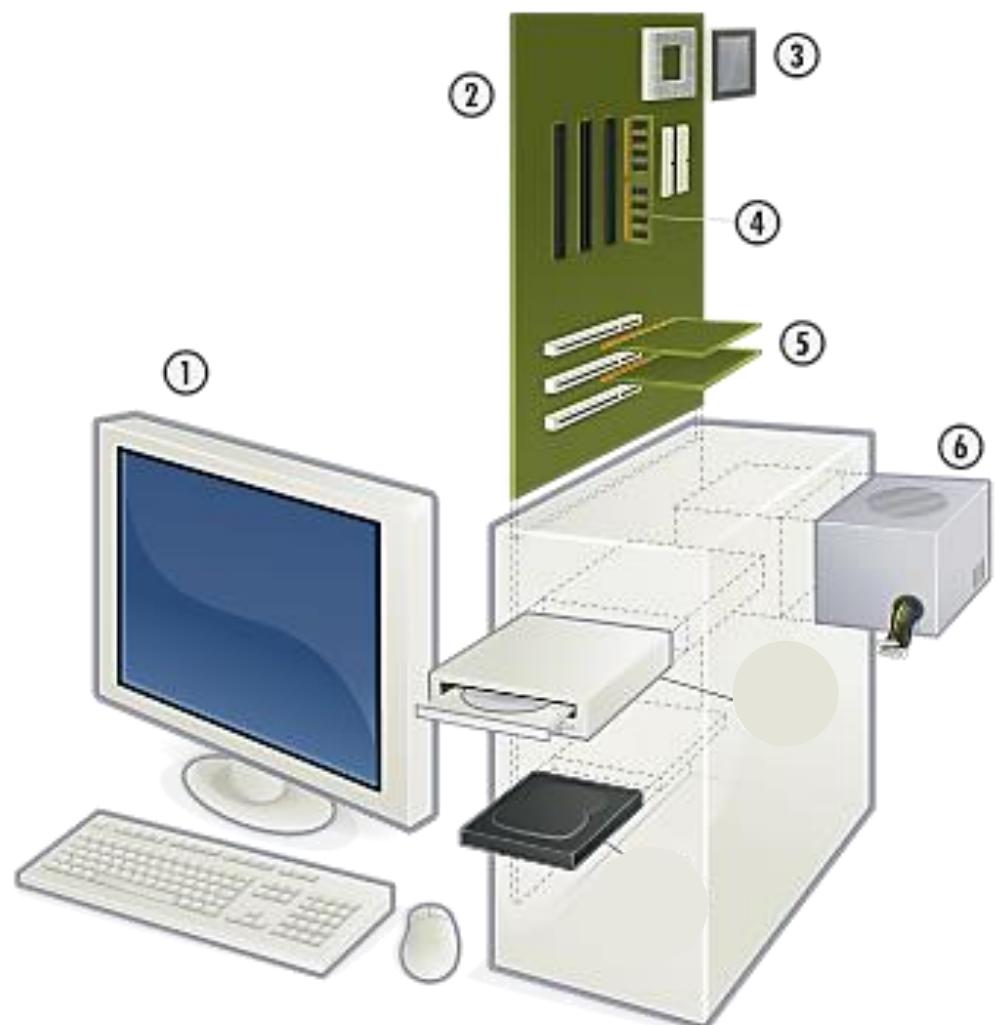
01 - Monitor

02 - Placa-Mãe

03 - Processador

04 - Memória RAM

05 - Placas de Rede, Som, Vídeo...



# Hardware - Review

01 - Monitor

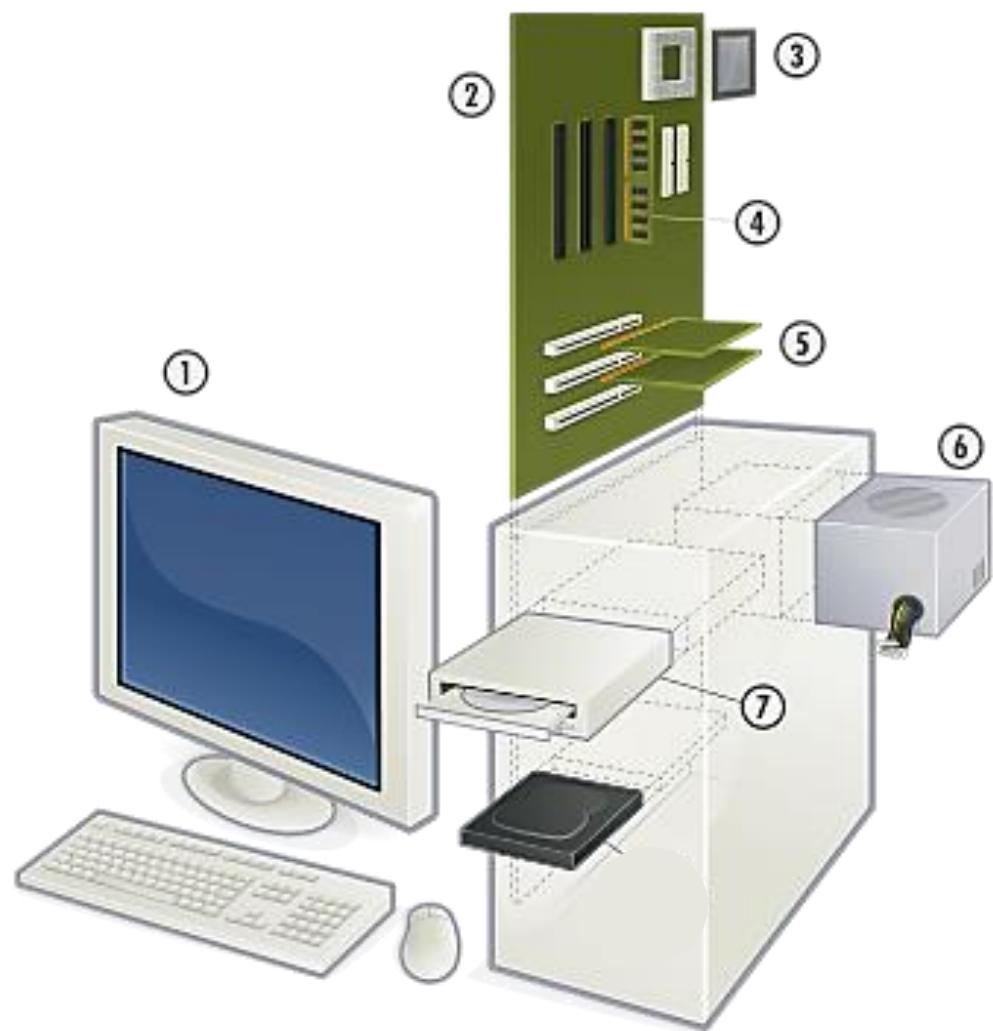
02 - Placa-Mãe

03 - Processador

04 - Memória RAM

05 - Placas de Rede,

06 - Fonte de Energia



# Hardware - Review

01 - Monitor

02 - Placa-Mãe

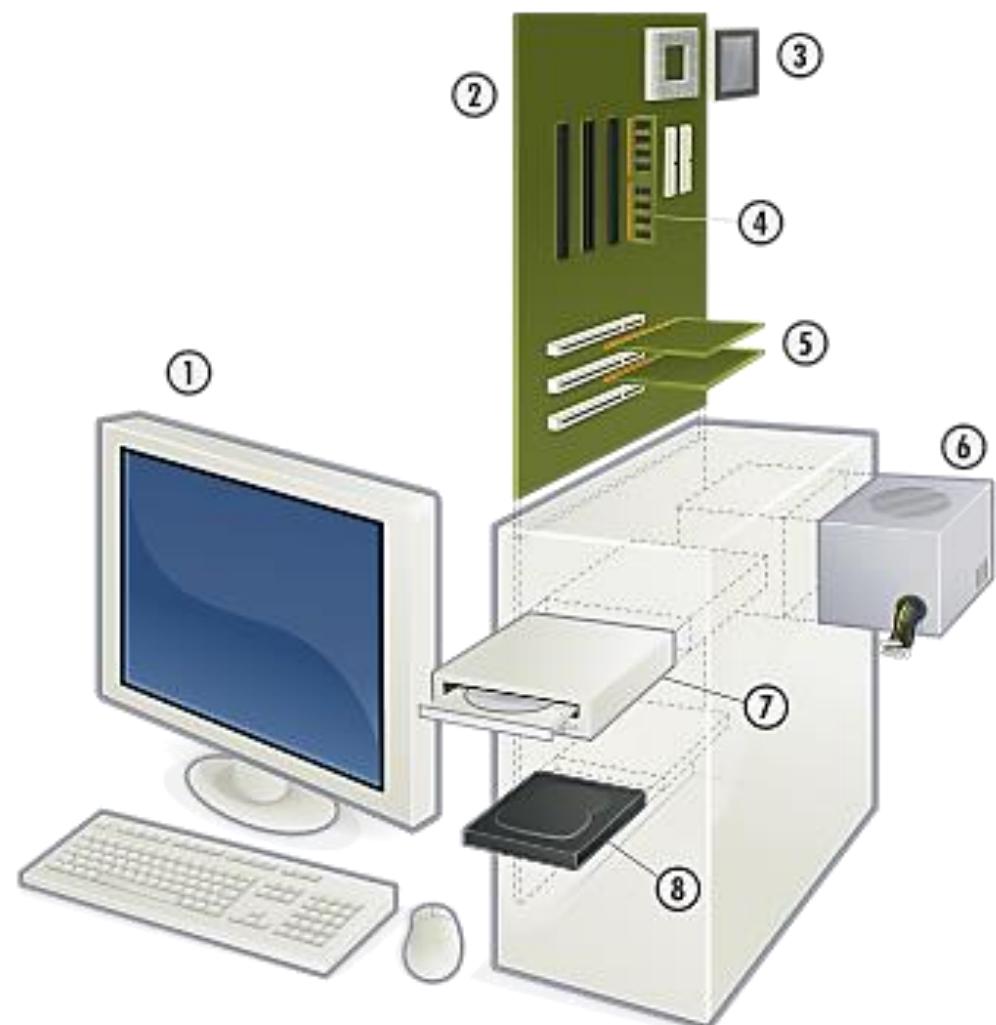
03 - Processador

04 - Memória RAM

05 - Placas de Rede, Som, Vídeo...

06 - Fonte de Energia

07 - Leitor de CDs e/ou DVDs



# Hardware - Review

01 - Monitor

02 - Placa-Mãe

03 - Processador

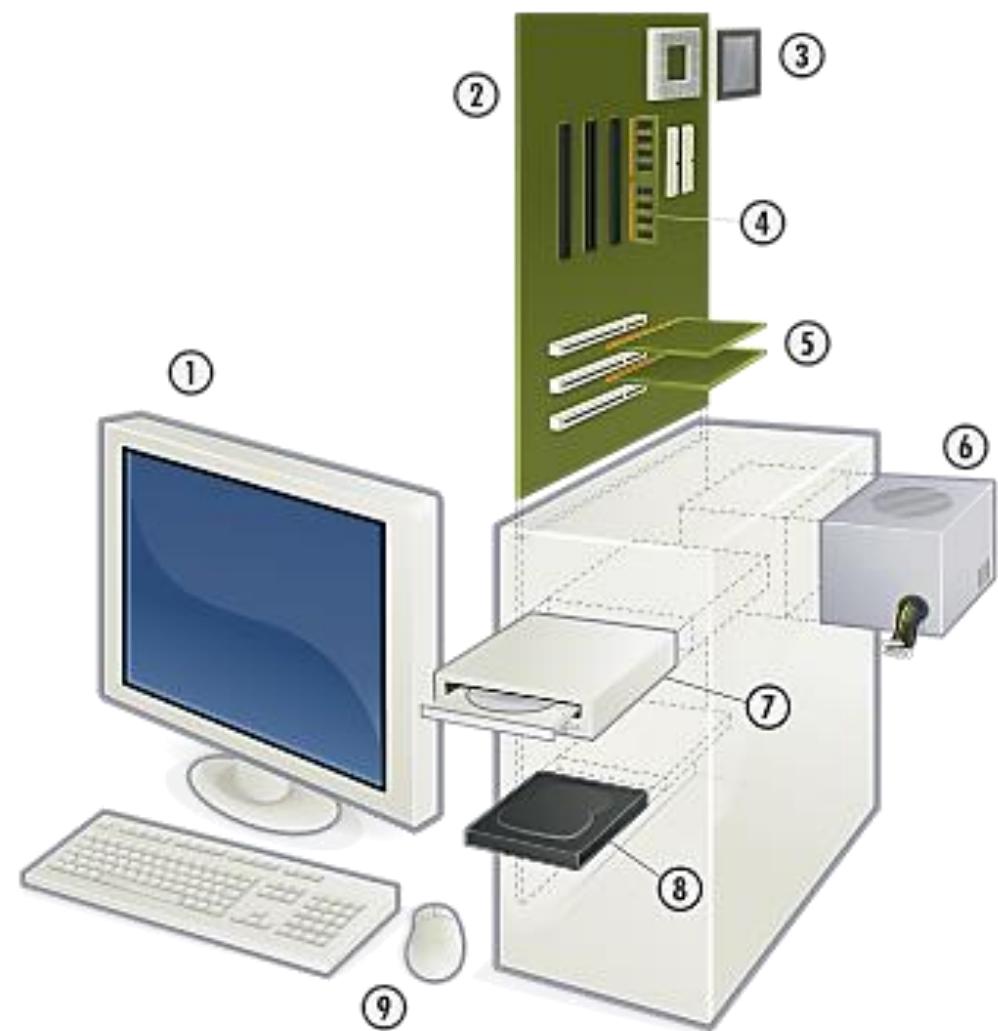
04 - Memória RAM

05 - Placas de Rede, Som, Vídeo...

06 - Fonte de Energia

07 - Leitor de CDs e/ou DVDs

08 - Disco Rígido (HD)



# Hardware - Review

01 - Monitor

02 - Placa-Mãe

03 - Processador

04 - Memória RAM

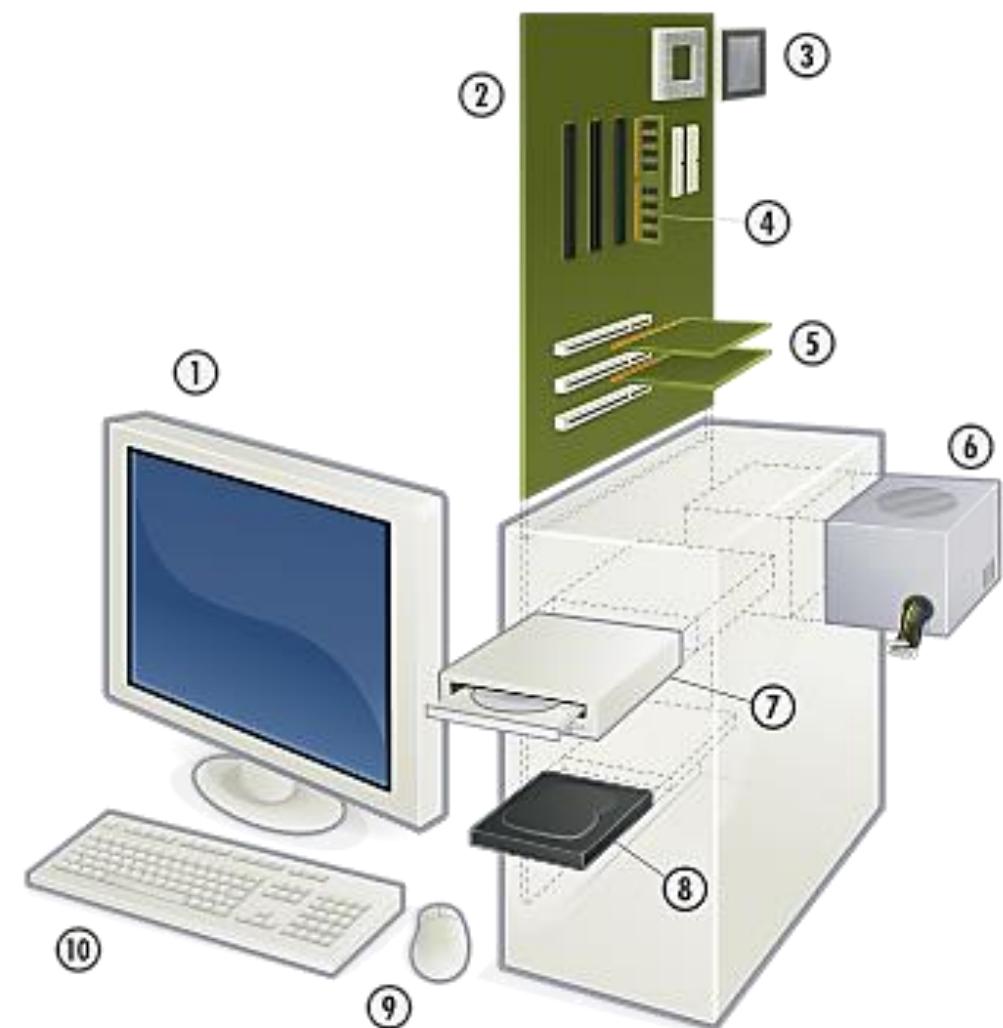
05 - Placas de Rede, Som, Vídeo...

06 - Fonte de Energia

07 - Leitor de CDs e/ou DVDs

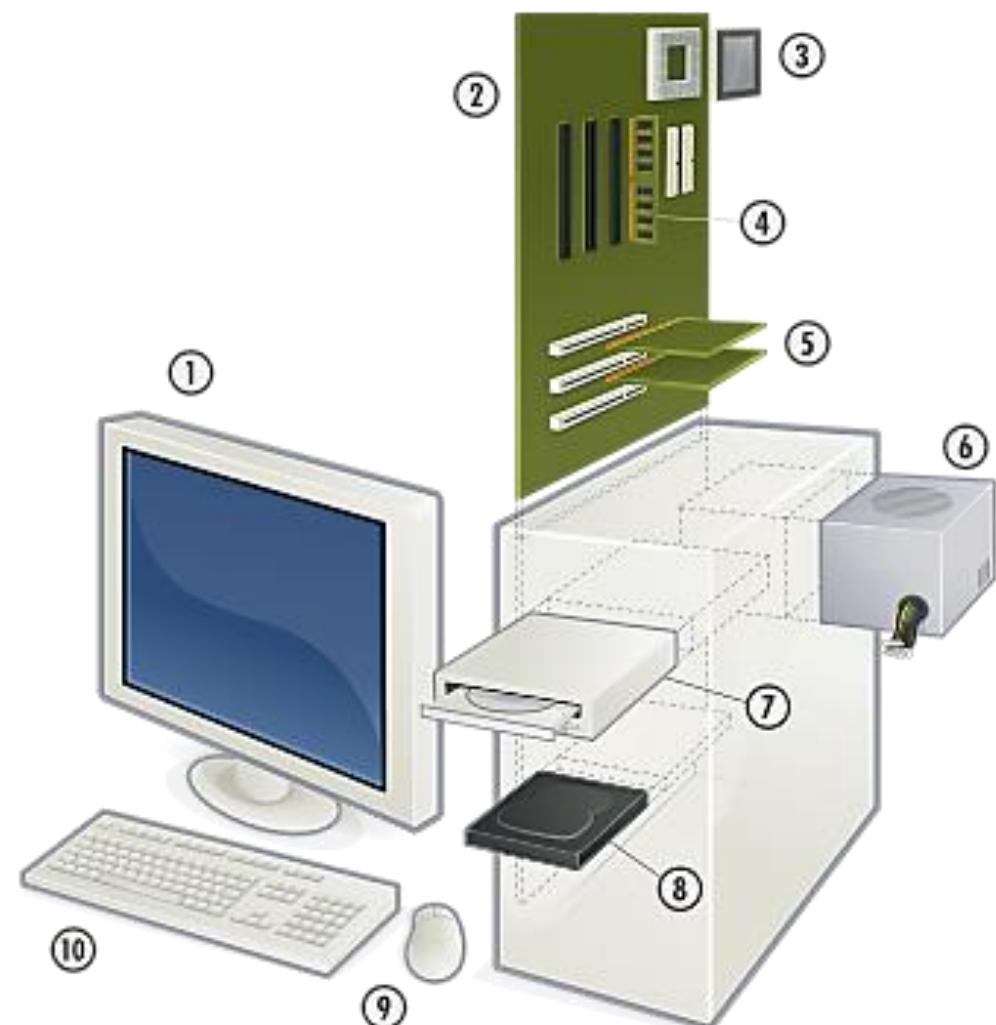
08 - Disco Rígido (HD)

09 - Mouse



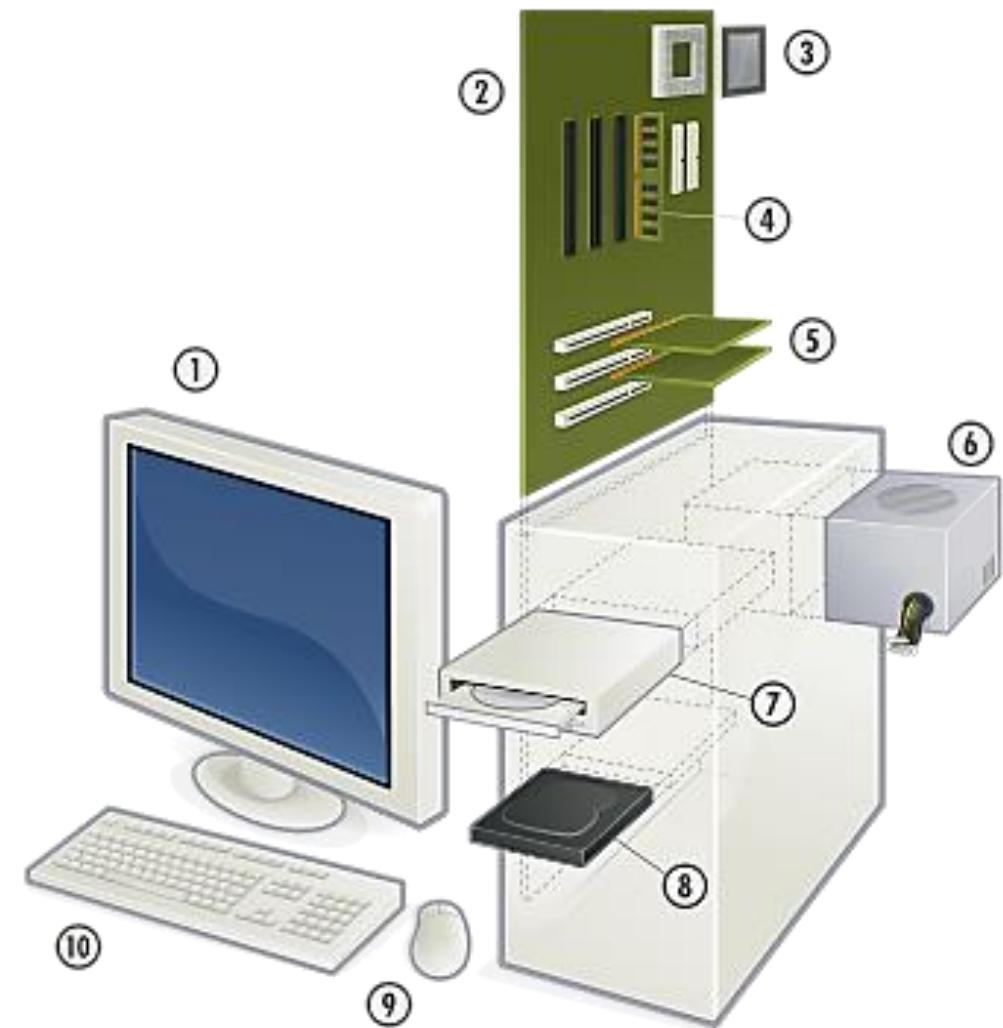
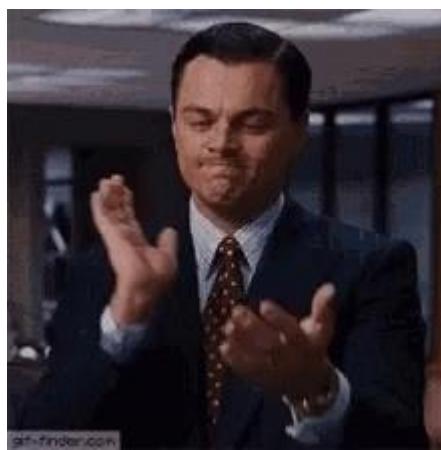
# Hardware - Review

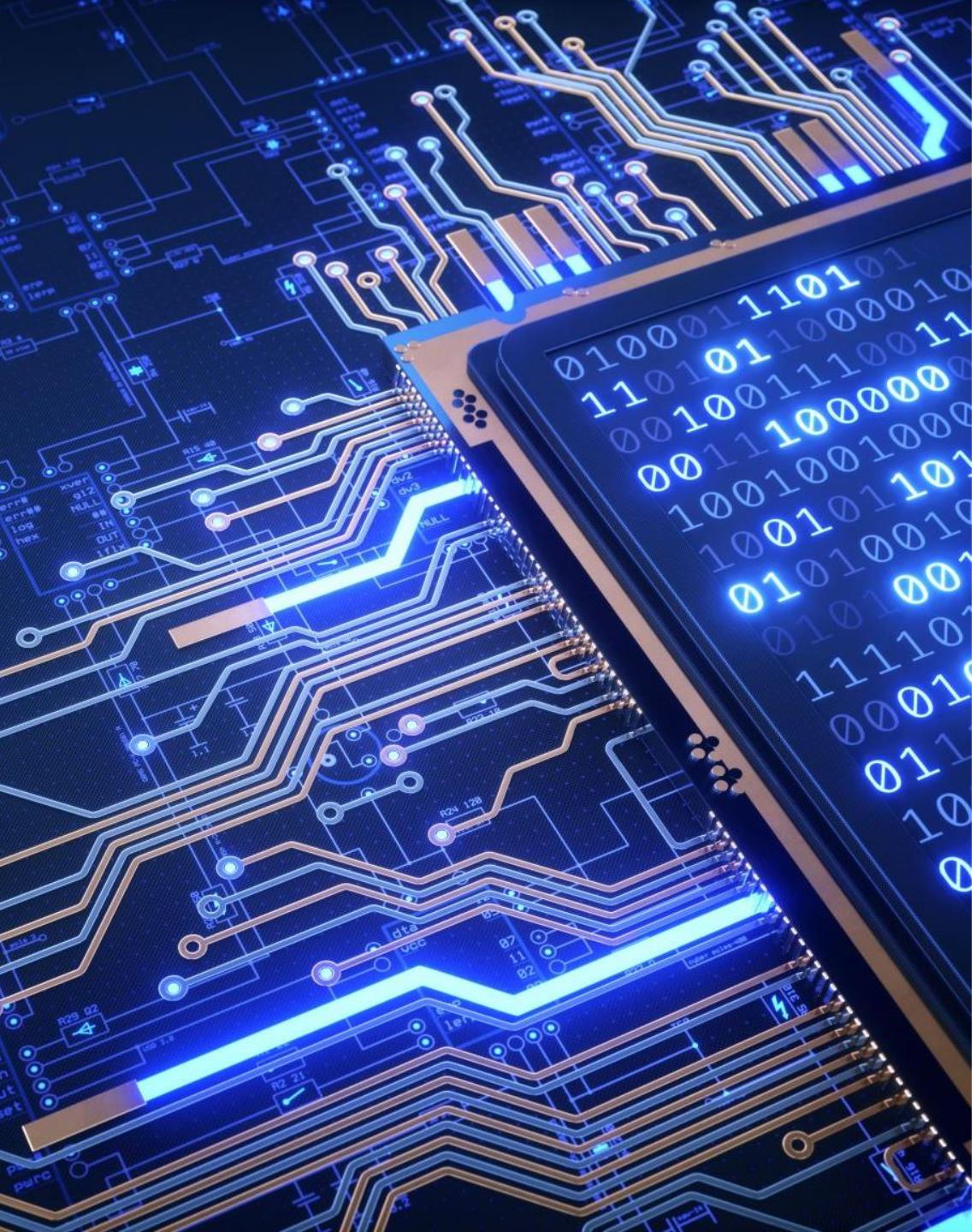
- 01 - Monitor
- 02 - Placa-Mãe
- 03 - Processador
- 04 - Memória RAM
- 05 - Placas de Rede, Som, Vídeo...
- 06 - Fonte de Energia
- 07 - Leitor de CDs e/ou DVDs
- 08 - Disco Rígido (HD)
- 09 - Mouse
- 10 - Teclado



# Hardware - Review

- 01 - Monitor
- 02 - Placa-Mãe
- 03 - Processador
- 04 - Memória RAM
- 05 - Placas de Rede, Som, Vídeo...
- 06 - Fonte de Energia
- 07 - Leitor de CDs e/ou DVDs
- 08 - Disco Rígido (HD)
- 09 - Mouse
- 10 - Teclado





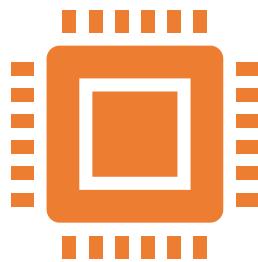
# Computador Digital: Introdução - Conceitos

## Computador Digital:

É uma máquina que pode resolver problemas executando uma série de instruções.

Máquina Programável.

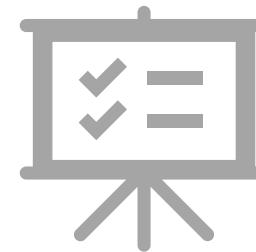
# Introdução - Conceitos



## Computador Digital:

É uma máquina que pode resolver problemas executando uma série de instruções.

Máquina Programável.



## Programa:

Uma sequência de instruções que descrevem a maneira de se realizar uma determinada tarefa.

# Introdução - Conceitos

## Linguagem de Máquina

Conjunto de instruções básicas que os circuitos eletrônicos de um determinado computador podem reconhecer e executar diretamente

### Linguagem primitiva (binária)

Somar 2 números

Comparar o valor de um número com zero.

Copiar um conjunto de dados de uma parte da memória para outra parte

“Complicada” para uso humano

# Linguagens de alto nível

ASP, ActionScript, C/C++, C#, Java, JavaScript, Lua, MATLAB, PHP, Python, R, Ruby, Visual Basic, ...

Facilitam a comunicação dos seres humanos com a máquina, sendo linguagens mais próximas dos humanos.

Devem ser convertidas em linguagem de máquina para serem executadas.

Métodos de conversão: a tradução e a interpretação.

# Tradução X Interpretação

## Tradução

- Programa de alto nível é primeiro convertido por completo para a linguagem de máquina para então ser executado.
- Em linguagens de alto nível, esta tradução é realizada pelo **compilador**
- (compilação ~ tradução)
- Ex: C, C++, C#.

## Interpretação

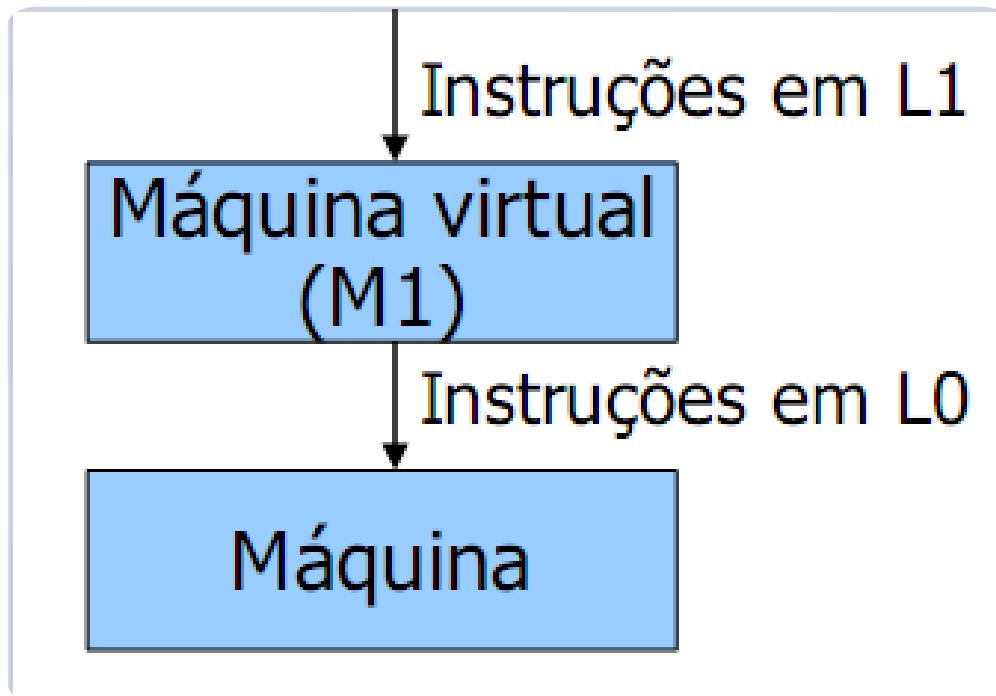
- Depois de cada instrução de alto nível ser examinada e decodificada, ela é executada imediatamente.
- Ex.: JavaScript .

## Métodos Híbridos

- Ex.: Java



# Máquina Virtual



Representa uma abstração capaz de reconhecer e executar diretamente as instruções de uma linguagem específica

---

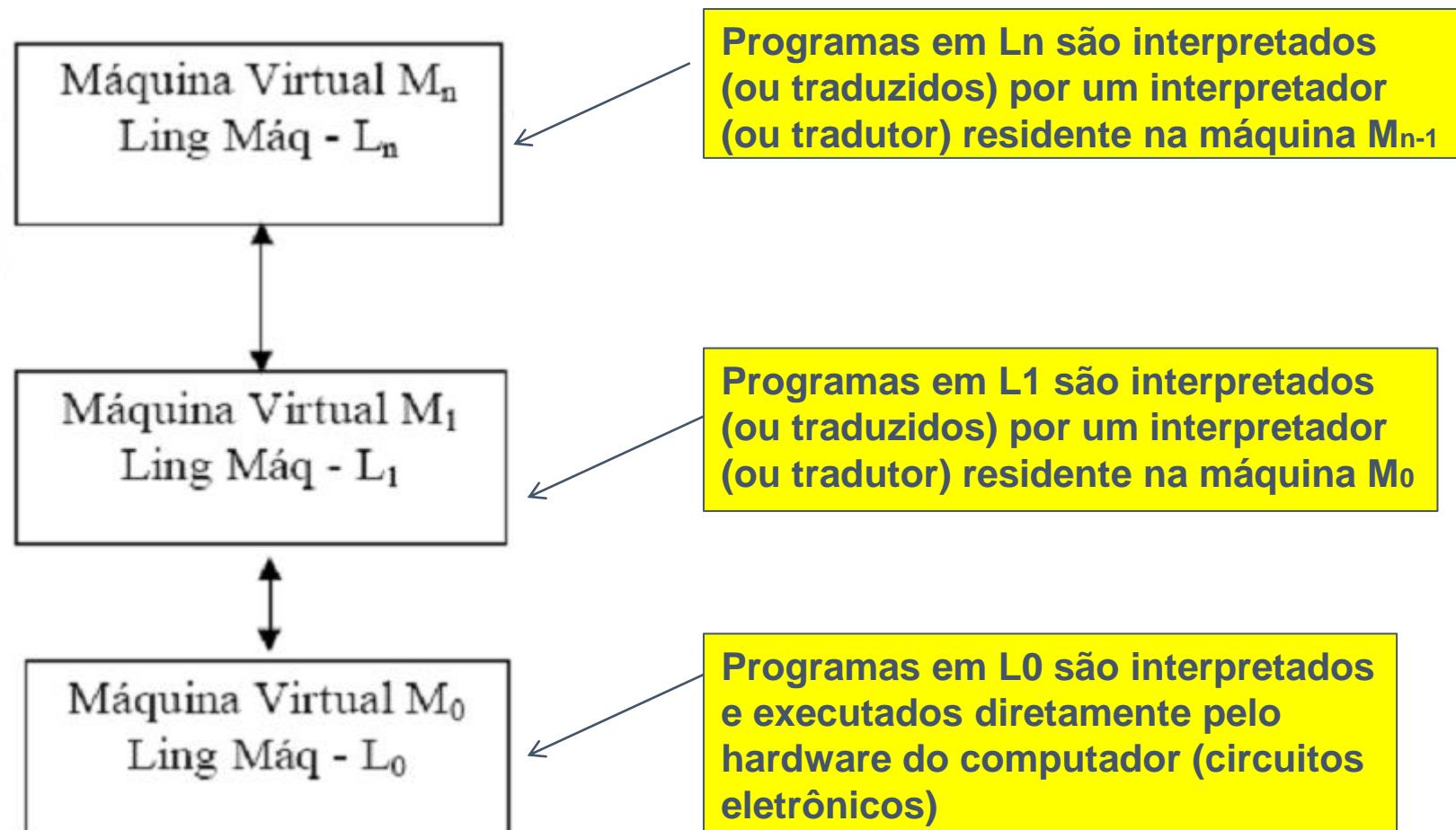
# Máquina Virtual

## Máquinas Multi-Níveis

Diferentes camadas ou níveis de abstração.



# Máquina de Vários Níveis



# Máquina de Vários Níveis

Cada máquina virtual tem associada a si uma linguagem, composta de todas as instruções que essa máquina pode executar.

**Uma máquina define uma linguagem.  
Uma linguagem define uma máquina.**

Um computador com  $n$  níveis pode ser visto como  $n$  máquinas virtuais distintas.

# Máquina de Vários Níveis

Vantagens desta abordagem:

Um computador pode ser considerado como composto por "n" máquinas virtuais, cada uma delas com sua própria configuração e linguagem de máquina;

# Máquina de Vários Níveis

Vantagens desta abordagem:

Um computador pode ser considerado como composto por "n" máquinas virtuais, cada uma delas com sua própria configuração e linguagem de máquina;

Uma pessoa, cujo trabalho seja gerar programas (aplicativos) para a máquina virtual de nível "k" ou outro qualquer, não precisa conhecer ou se preocupar com as particularidades dos níveis inferiores ao seu;

# Máquina de Vários Níveis

## Vantagens desta abordagem:

Um computador pode ser considerado como composto por "n" máquinas virtuais, cada uma delas com sua própria configuração e linguagem de máquina;

Uma pessoa, cujo trabalho seja gerar programas (aplicativos) para a máquina virtual de nível "k" ou outro qualquer, não precisa conhecer ou se preocupar com as particularidades dos níveis inferiores ao seu;

A estrutura de níveis permite ver o computador como um conjunto hierárquico de facilidades que possibilitem flexibilidade e independência ao usuário.

# Máquina de Vários Níveis



Nível de linguagem orientada para problemas

Tradução (compilador)

Nível de linguagem de montagem

Tradução (montador)

Nível de sistema operacional

Interpretação parcial (sistemas)

Nível convencional de máquina

Interpretação (microprogramas ou execução direta)

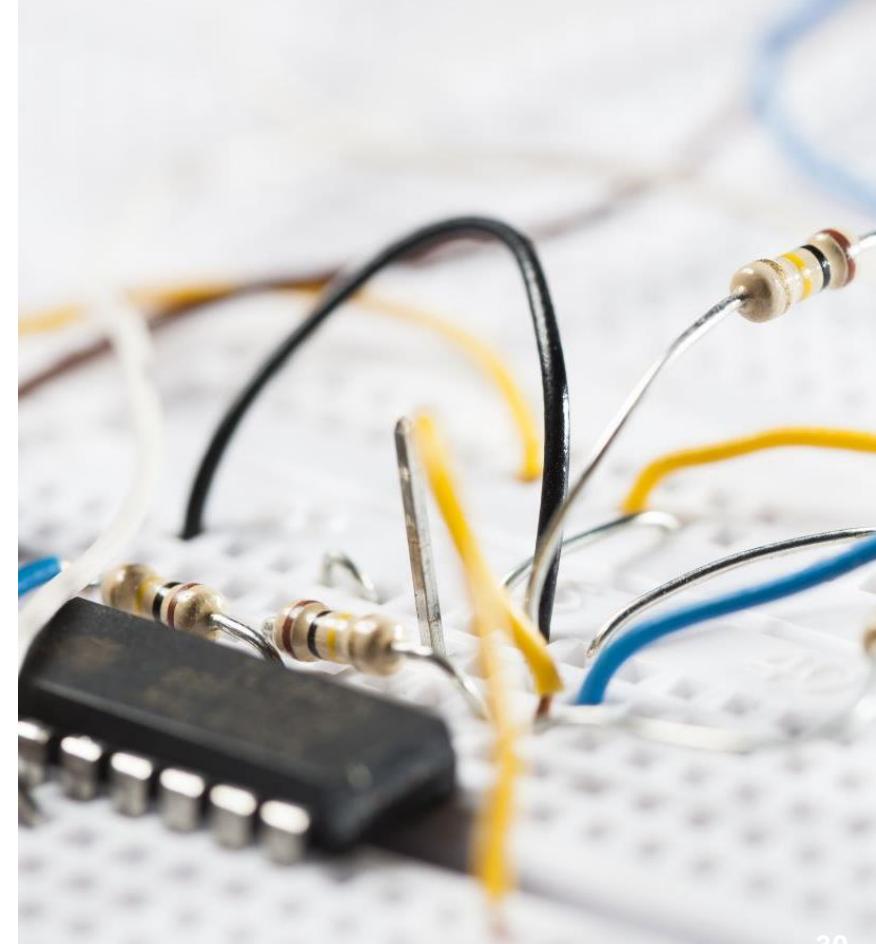
Nível de microprogramação ou microarquitetura

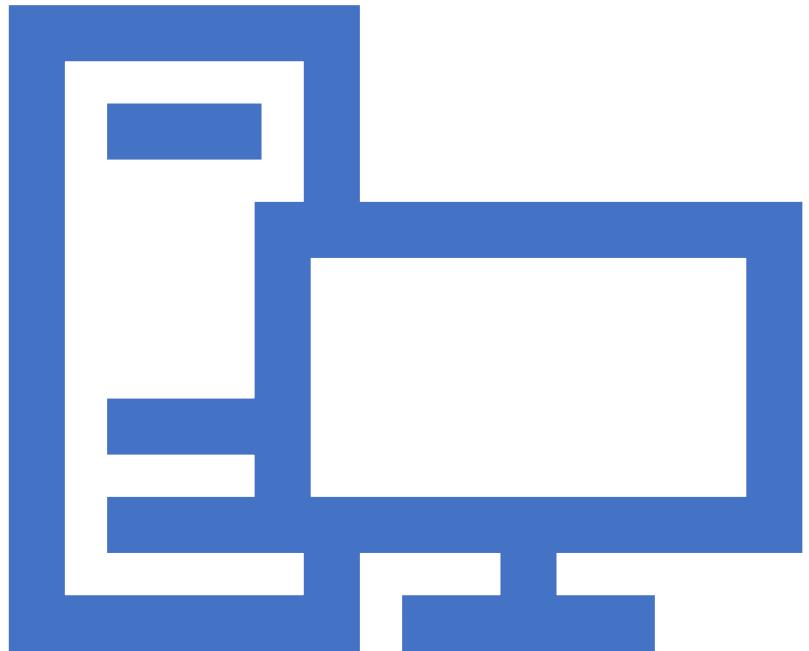
Os microprogramas são executados diretamente pelo hardware

Nível de lógica digital

# Nível dos Dispositivos

Situado abaixo do nível 0  
Microeletrônica  
Características físicas  
Malha de transistores  
Tecnologias de fabricação de circuitos integrados

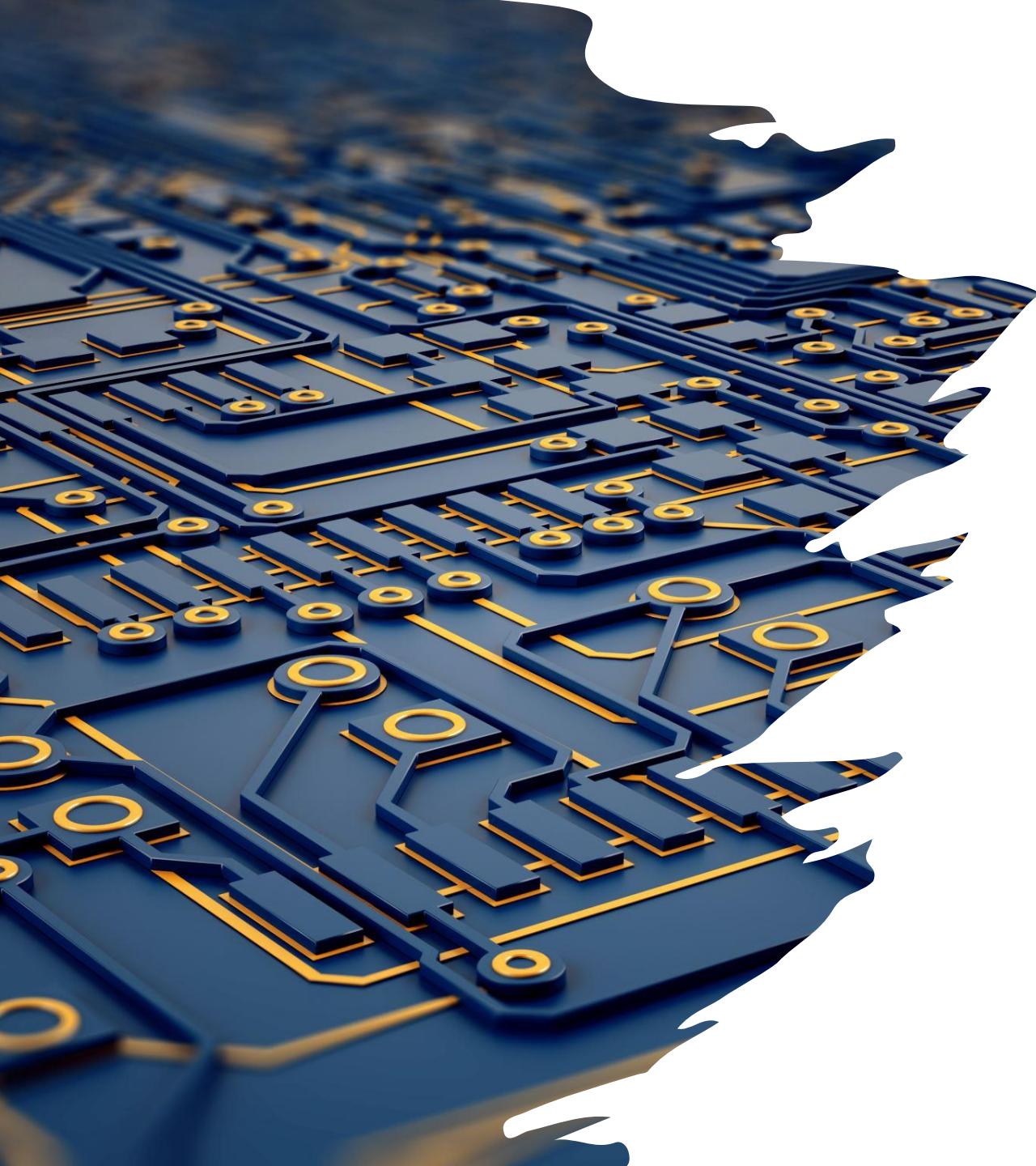




## Nível 0 ou Nível da Lógica Digital

É composto pelo hardware da máquina

O nível 0 executa diretamente as instruções (ou microinstruções) submetidas pelo nível 1.



# Nível 1 ou Nível da Microarquitetura

Nesse nível, inicia-se o conceito de programa como uma sequência de instruções a serem executadas diretamente pelos circuitos eletrônicos.

“Enxerga-se”:

- Um conjunto de registradores (memória local)
- Um circuito chamado **ULA (Unidade Lógica e Aritmética)**

# Nível 1 ou Nível da Microarquitetura

Em algumas arquiteturas, no nível 1 há um programa denominado microprograma

**Função:** interpretar as instruções de nível 2, executando-as em seguida

É o microprograma quem controla a operação do caminho de dados

# Nível 1 ou Nível da Microarquitetura

## Controle por software

- Microprograma = Interpretador (residente em memória ROM)
  - Busca, decodifica e executa as instruções, uma a uma, usando o caminho de dados para a realização de uma tarefa.

## Controle por hardware

- Os mesmos passos acima são executados, mas sem que haja um programa armazenado para controlar a interpretação das instruções

# Nível 1 ou Nível da Microarquitetura

## Vantagens da microprogramação

Facilitar o projeto e a construção dos circuitos digitais

- Implementação de parte da lógica digital dentro do firmware

Flexibilizar e possibilitar o desenvolvimento de instruções mais potentes a nível de máquina convencional.

## Nível 2 ou Nível ISA

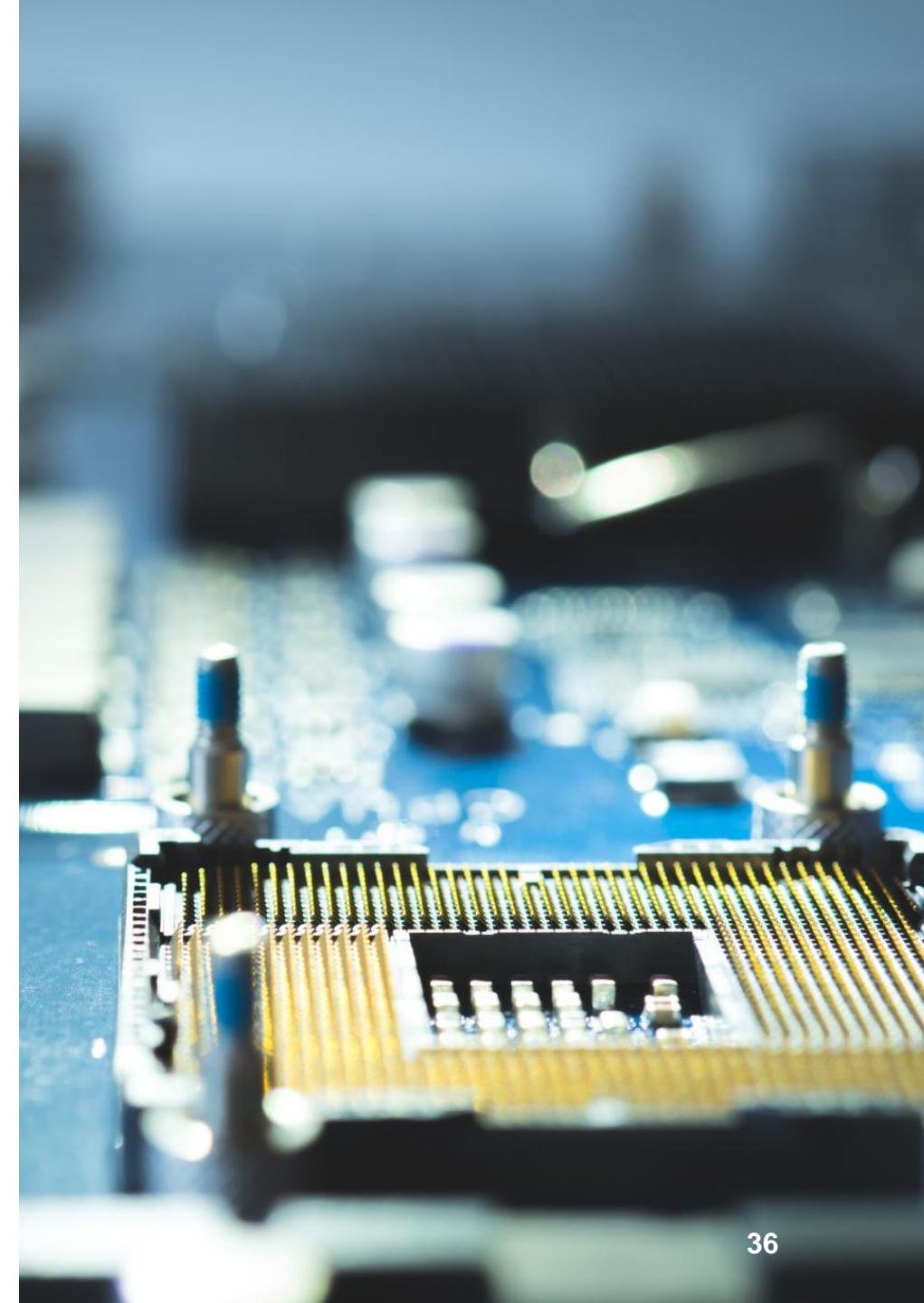
### Nível ISA – *Instruction Set Architecture*

- Ou Nível da Arquitetura do Conjunto de Instruções

### Nível Convencional de Máquina

Define o conjunto das Instruções executáveis por uma máquina (processador)

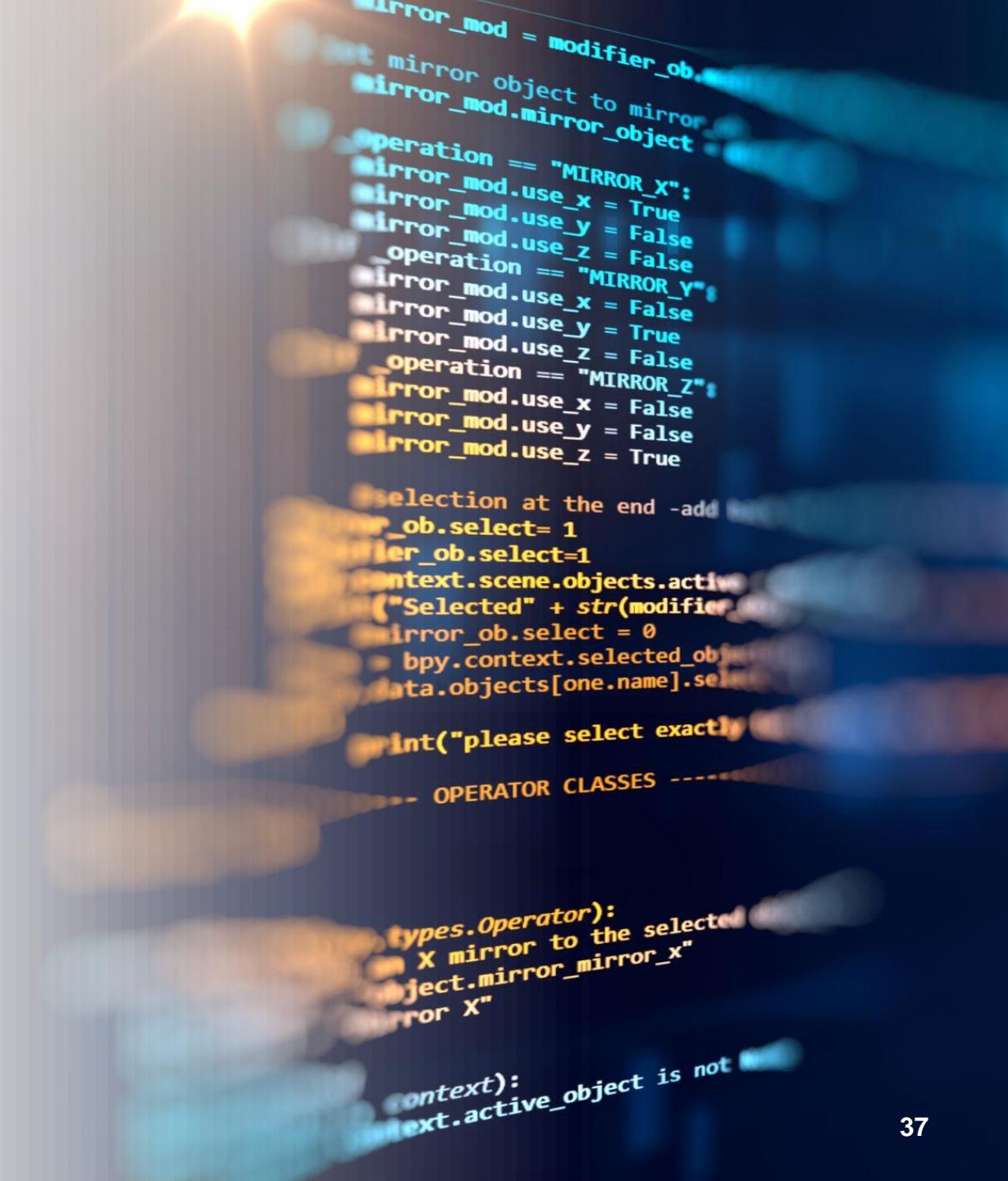
- Cada máquina ou processador tem sua linguagem própria de nível 2, chamada “**Linguagem de Máquina**” (documentada em manuais específicos de cada fabricante).



## Nível 3 ou Nível do Sistema Operacional (SO)

Fornece serviços básicos para os níveis acima

- Interface (gráfica ou linha de comando) com o usuário
- Gerenciamento de memória
- Escalonamento de processos
- Acionamento de dispositivos de entrada e saída de dados, etc.



```
mirror_mod = modifier_ob
# mirror object to mirror
mirror_mod.mirror_object = mirror_object
if mirror_mod.mirror_object:
    if mirror_mod.mirror_object.type == 'MESH':
        if mirror_mod.mirror_object.mode == 'OBJECT':
            if mirror_mod.mirror_object.select == 1:
                if mirror_mod.mirror_object.name == modifier.name:
                    if mirror_mod.mirror_use_x == True:
                        if mirror_mod.mirror_use_y == False:
                            if mirror_mod.mirror_use_z == False:
                                if mirror_mod.mirror_use_x == False:
                                    if mirror_mod.mirror_use_y == True:
                                        if mirror_mod.mirror_use_z == False:
                                            if mirror_mod.mirror_use_x == False:
                                                if mirror_mod.mirror_use_y == False:
                                                    if mirror_mod.mirror_use_z == True:
                                                        if selection at the end -add
                                                        mirror_mod.select= 1
                                                        modifier.select=1
                                                        bpy.context.scene.objects.active = modifier
                                                        ("Selected" + str(modifier))
                                                        mirror_mod.select = 0
                                                        bpy.context.selected_objects = []
                                                        data.objects[one.name].select = 1
                                                        print("please select exactly one object")
                                                        - OPERATOR CLASSES -
types.Operator):
    X mirror to the selected object.mirror_mirror_x"
    or X"
context):
    context.active_object is not None
```

# Nível 3 ou Nível do Sistema Operacional (SO)

**Níveis abaixo:** dirigidos aos **programadores de sistema**

**Níveis acima:** dirigidos aos **programadores de aplicação**

# Nível 4 ou Nível de Linguagem de Montagem



Provê um método para as pessoas escreverem programas para os níveis 1, 2, e 3 de uma maneira não tão desconfortável. Mnemônicos para as instruções de máquina.



Os programas escritos em linguagem de montagem são primeiramente traduzidos para a linguagem dos níveis 1, 2 ou 3, e depois interpretados.



**Montador:** programa que executa a tradução dos programas em linguagem de montagem para uma linguagem do nível 1, 2 ou 3.

## Nível 5 ou Nível de Linguagens de Alto Nível

Linguagens projetadas para serem utilizadas por programadores de aplicação com problemas a serem resolvidos.

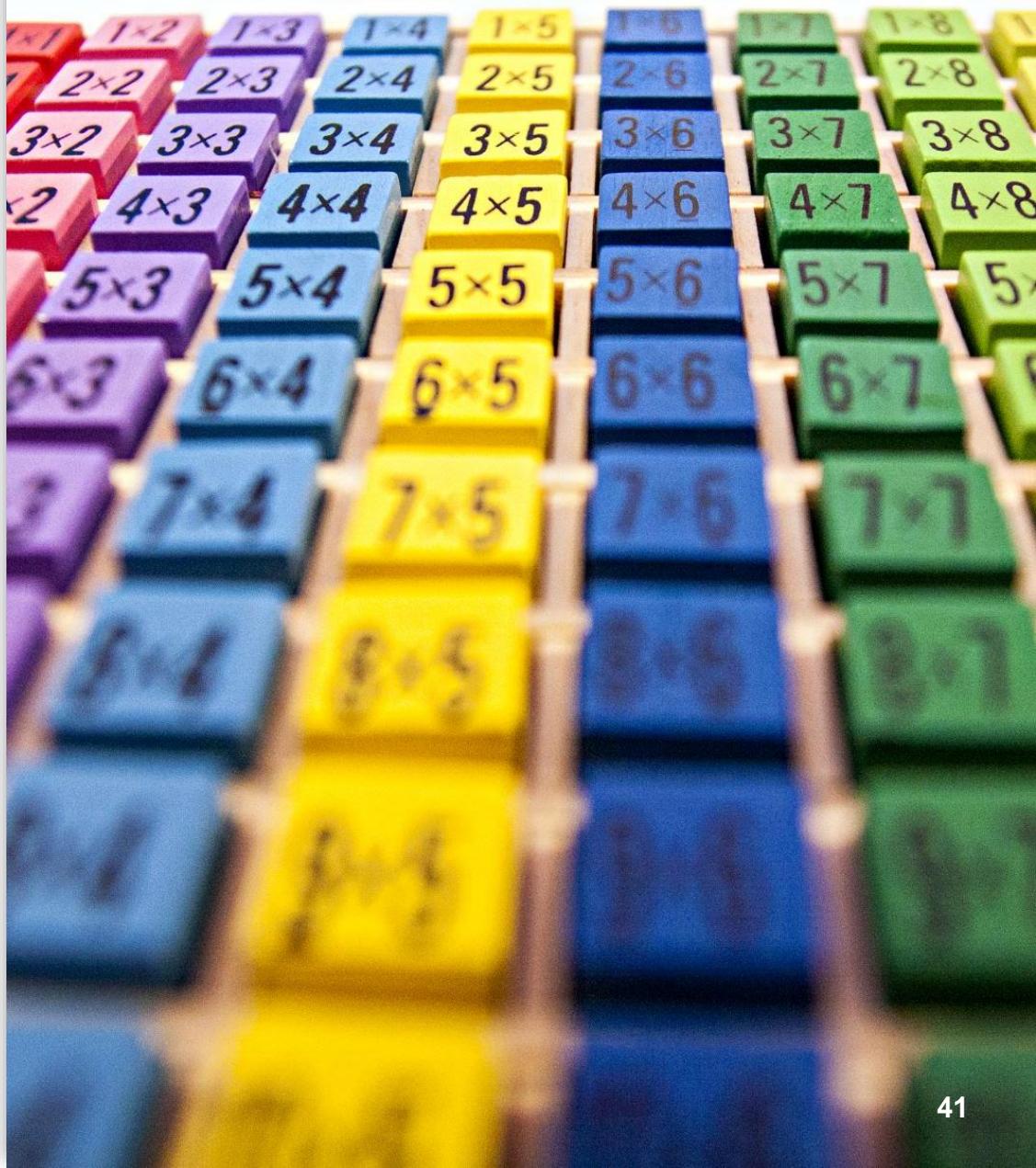
Ex. de linguagens de alto nível: C, C++, Java...

# Mais Níveis?

Acima do nível 5 encontram-se coleções de programas projetados para criar máquinas especialmente adequadas para certas aplicações (ou domínios), contendo grandes quantidades de informação acerca da aplicação.

Máquinas virtuais voltadas às aplicações

- Administração, educação, projeto de computadores, realidade virtual, etc.



# Arquitetura de Computadores

## Pontos fundamentais:

Computadores são projetados como uma série de níveis

Cada nível é construído em cima de seus precursores.

Cada nível representa uma abstração distinta, com diferentes objetos e operações presentes

**Arquitetura do Nível:** conjunto de tipos de dados, instruções e características

Abstrai-se o que é irrelevante, reduz a complexidade e foca no que interessa.

**Arquitetura de Computadores:** é o estudo de como projetar as partes de um sistema de computador visíveis aos programadores.

# Hardware, Software e Firmware

**Hardware**

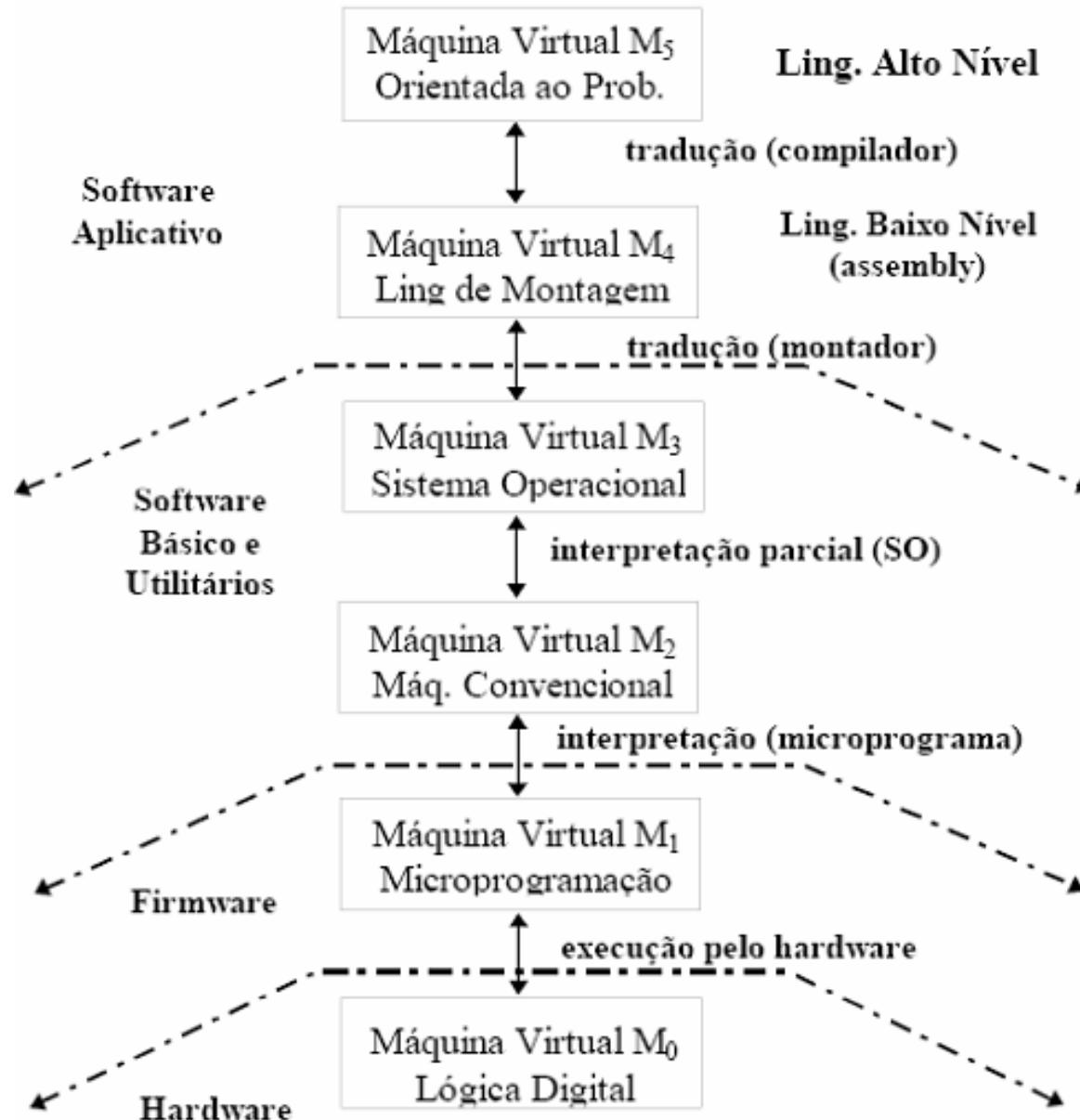
**Software**

**Hardware e Software são logicamente equivalentes**

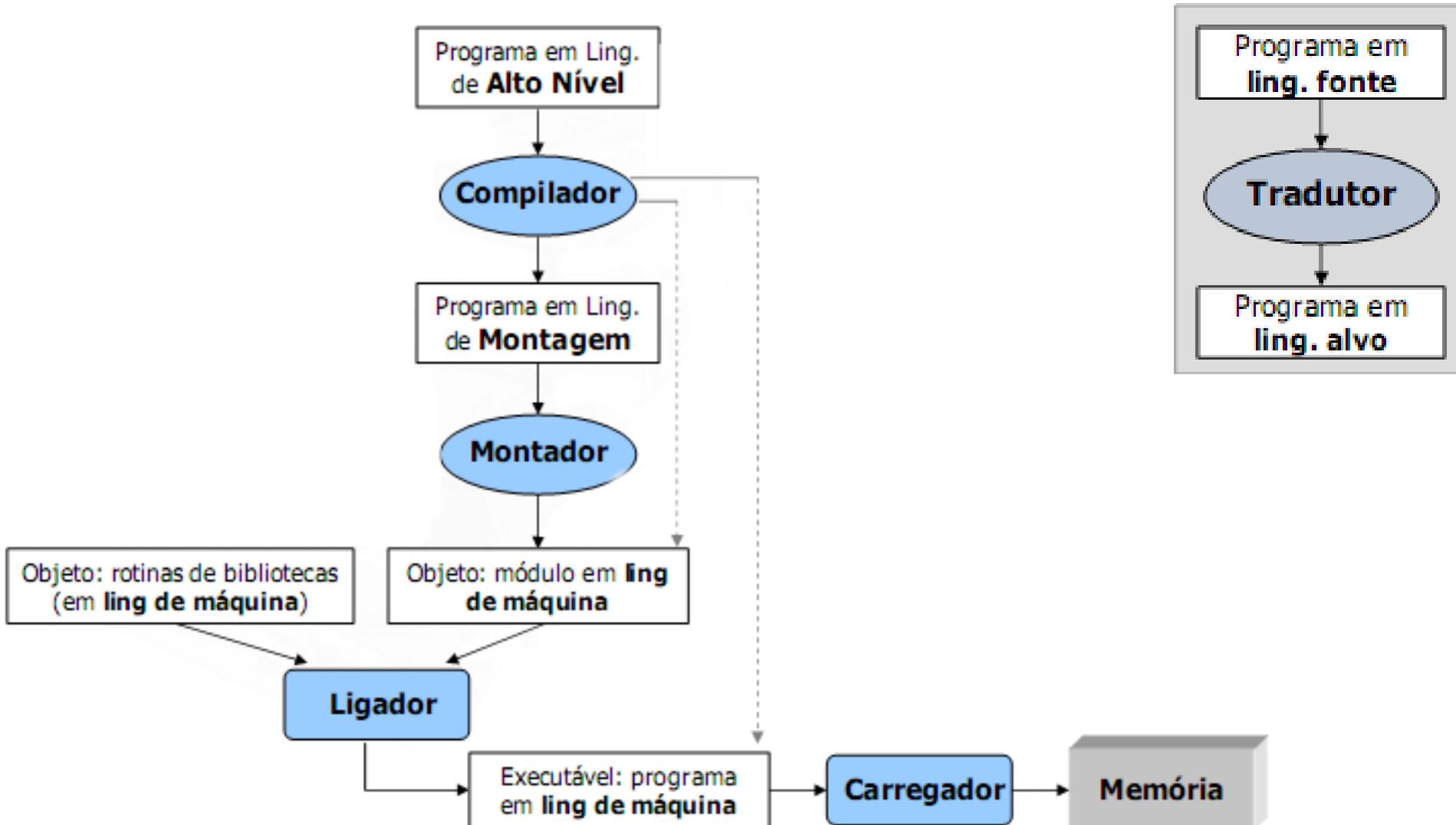
**Firmware**



# Hardware, Software e Firmware



# Execução de um Programa



# Compiladores, Montadores, Ligadores e Carregadores

## Compiladores

Ambos são TRADUTORES!

## Montadores (*Assemblers*)

# Compiladores, Montadores, Ligadores e Carregadores

**Ligadores (Linkers)**

**Carregadores (Loaders)**

# Interpretadores

Recebem como entrada arquivos texto contendo programas em linguagem assembly, ou linguagem de alto nível, ou arquivos binários com instruções de máquina, e os executam diretamente.

# RISC X CISC

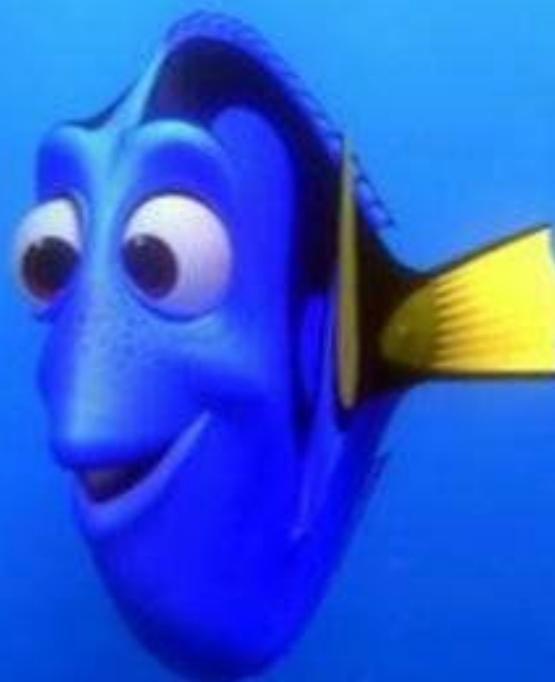
**CISC – *Complex Instruction Set Computer***

**RISC – *Reduced Instruction Set Computer***

**Tamanho do código X Desempenho**

Memórias

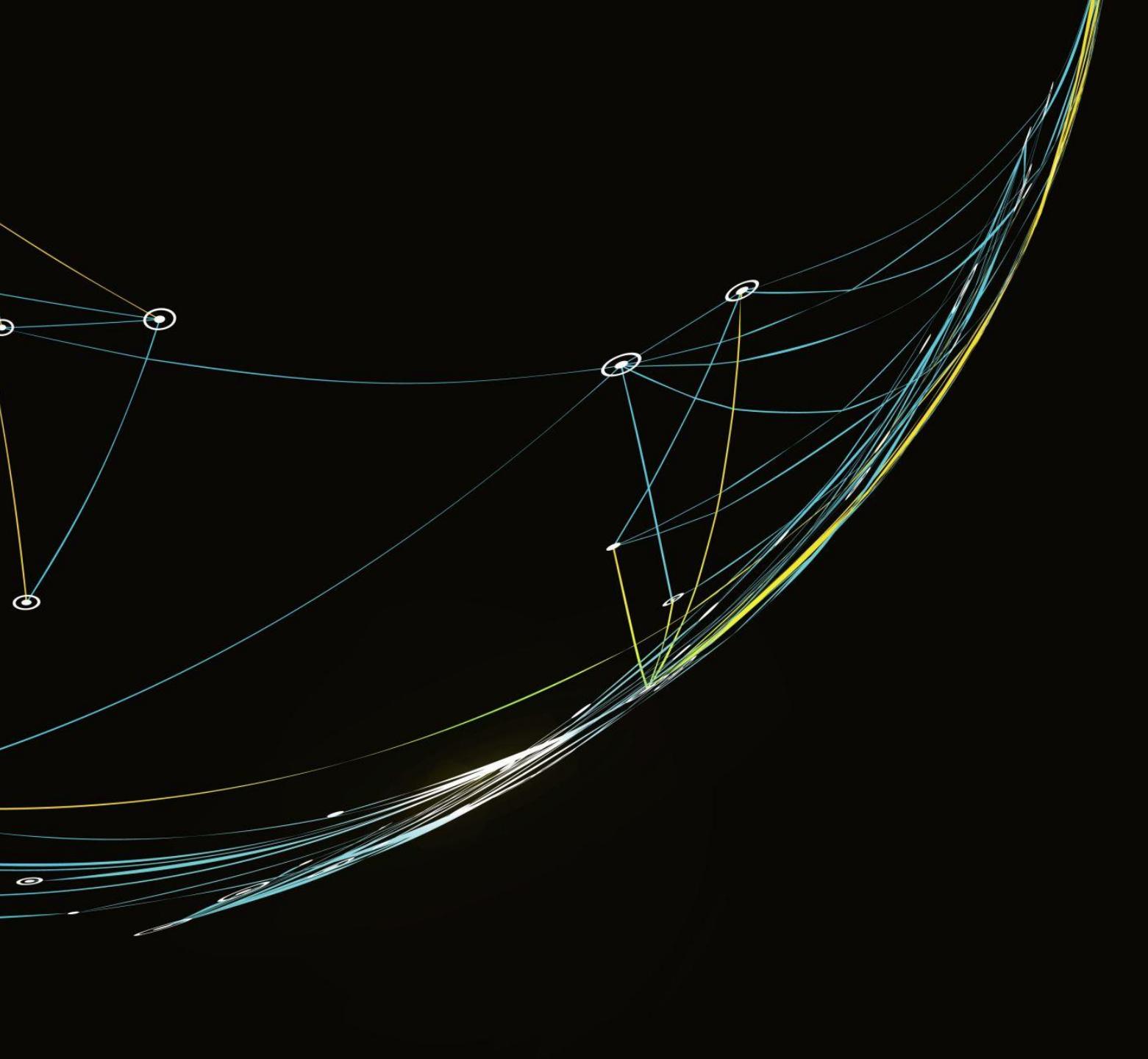
I MAY HAVE BAD MEMORY



BUT AT LEAST I DON'T  
HAVE BAD MEMORY

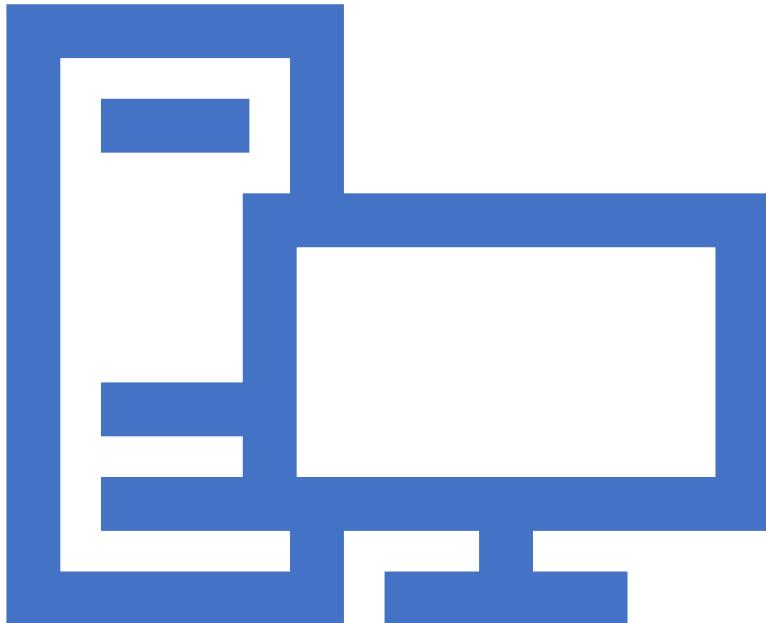
# MEMÓRIAS





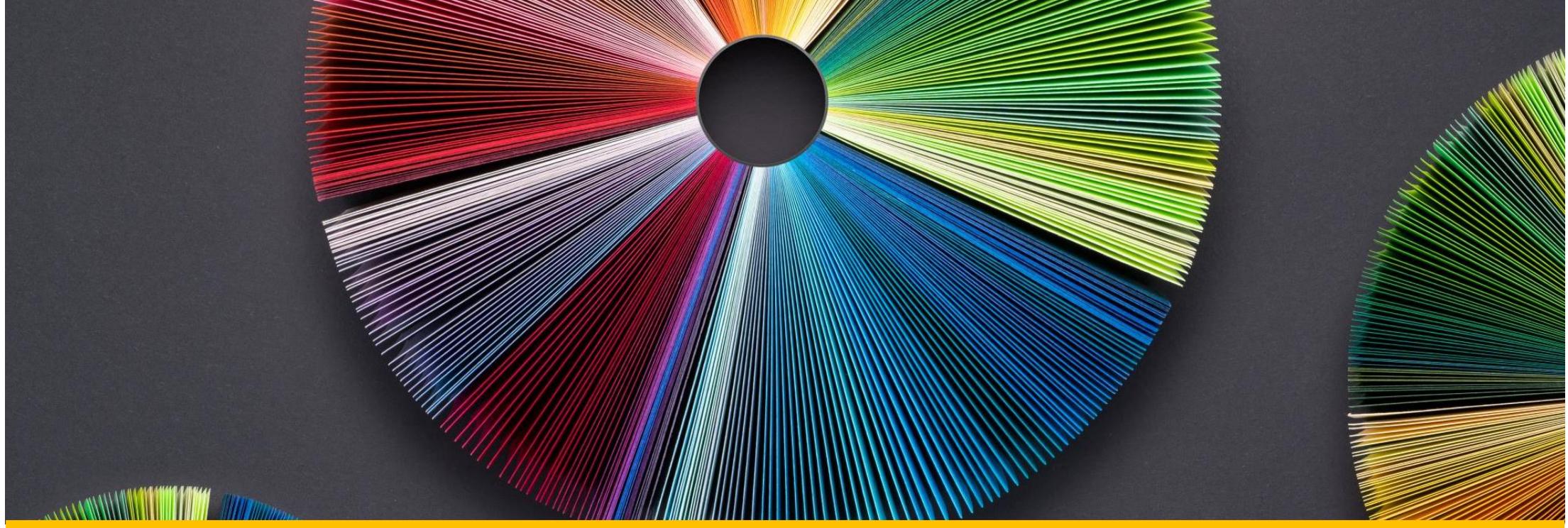
## Memórias

- **Função:** Armazenar dados e programas
- Para serem executados pela CPU, os programas devem estar armazenados em memória



## Memória Principal

- É a memória de trabalho, nela são carregados os programas e dados usados pela CPU no momento.
- Geralmente seu conteúdo é perdido uma vez que o computador seja desligado.  
Ex.: RAM (*Random Access Memory*)

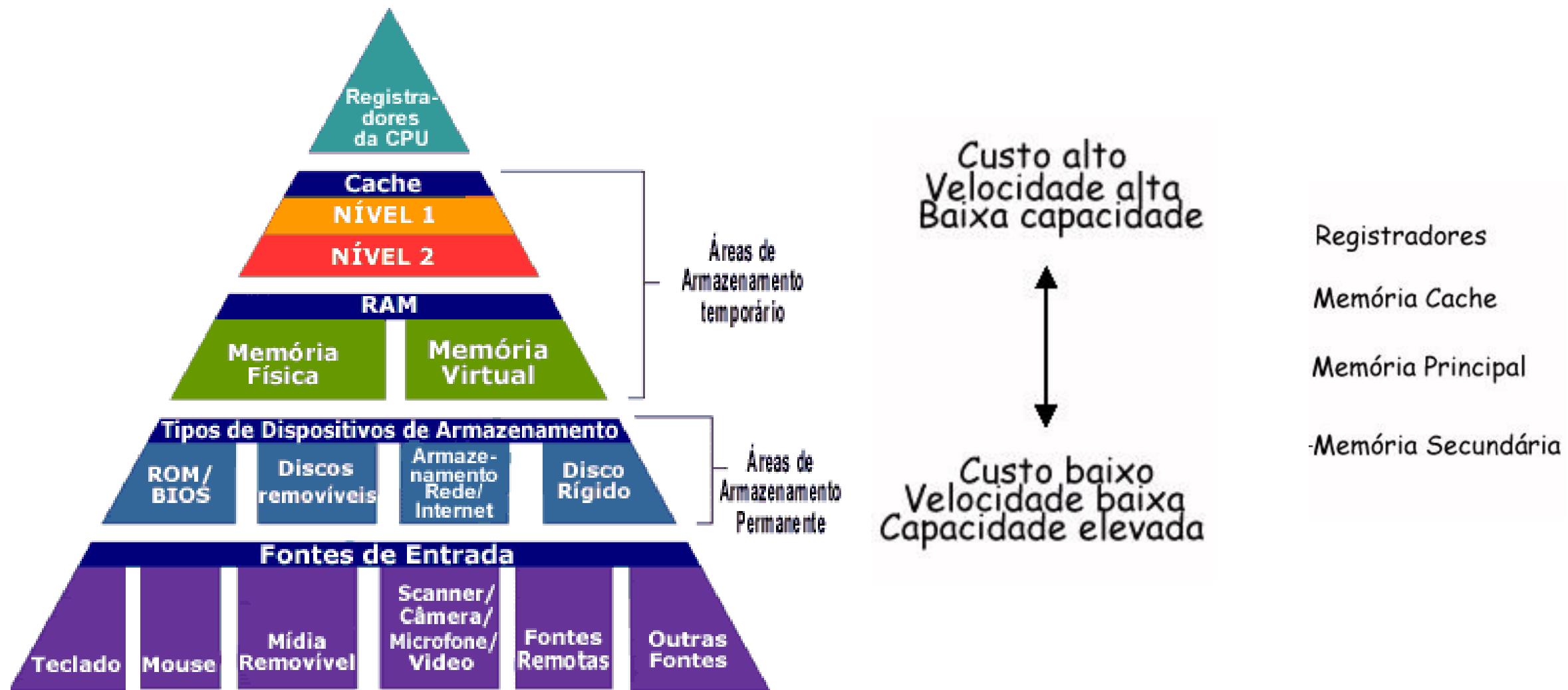


## Memória secundária

- Grande capacidade de armazenamento
- Mais lentas do que a MP
- Geralmente não-voláteis, permitindo guardar dados permanentemente.
- Ex.: Discos rígidos (HD - Hard Disk), CDs, DVDs, SSDs, Pendrives

E a Memória Cache?

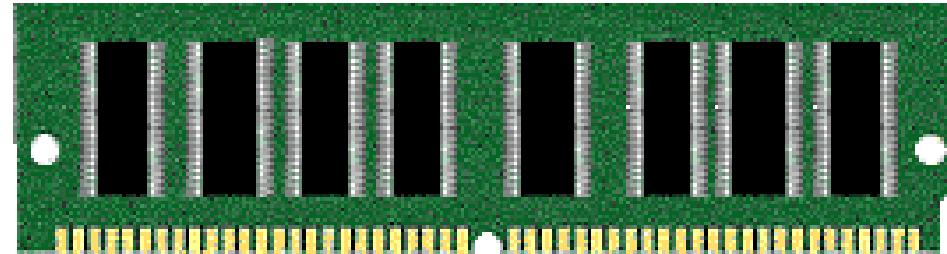
# Memórias



# SIMMs - Single In-line Memory Modules (Módulos de memória simples em linha)

Estilo mais antigo de pacote para RAM.

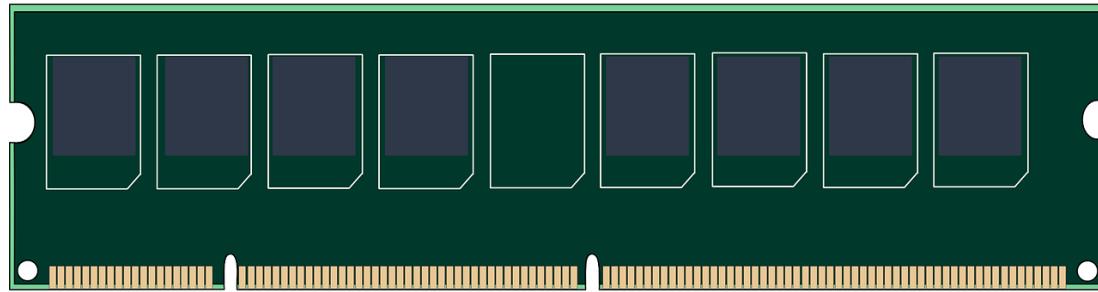
**Obs.:** As memórias SIMM foram os primeiros pentes de memória a serem produzidos em massa.



**Curiosidade:** Antes das memórias SIMM, a maioria das memórias eram instaladas diretamente na *motherboard* (não havia muita possibilidade de *upgrade*).

# DIMMs - Dual Inline Memory Module (módulo de memória dupla em linha)

Apresentam 168 pinos e oferecem largura de banda de 64 bits  
Têm 13,65 cm (5,375 pol) de comprimento e 3,81 cm (1,5 pol) de altura



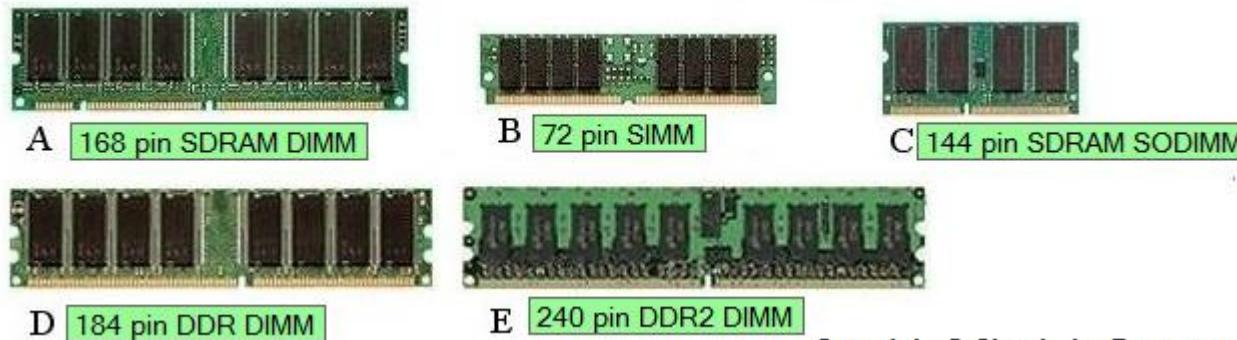
# SDRAM - Synchronous Dynamic RAM (RAM dinâmica síncrona)

Disponível em pacotes SIMM e DIMM, e em duas configurações:

RAM sem paridade (também conhecida como RAM não ECC)  
Mais barata que a RAM com paridade.

RAM com paridade (também conhecida como RAM ECC)  
Contém circuitos extras que podem ajudar a minimizar erros específicos de RAM.

Obs.: A RAM com paridade e a RAM sem paridade normalmente não são compatíveis entre si.



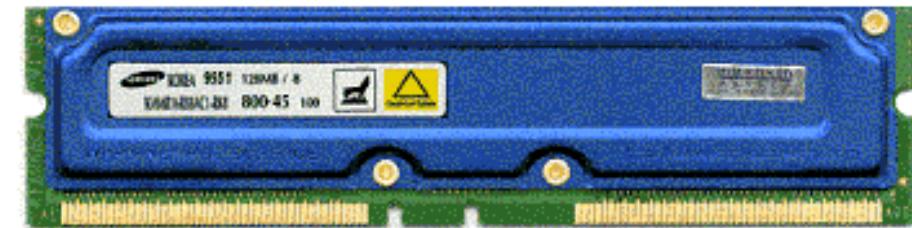
# DDR SDRAM - Double Data Rate-Synchronous DRAM (DRAM síncrona com taxa de transferência de dados dupla)

Permite a transferência do dobro de memória por ciclo.  
Também pode ser chamada de SDRAM II or DDRAM.

# RIMM (RDRAM e CRIMM)

RIMM - Rambus Inline Memory Module (Módulo de memória em linha Rambus)

Também conhecida como RDRAM



**Obs.:** Se apenas um RIMM estiver sendo usado, os outros slots precisarão ser preenchidos com um CRIMM

CRIMM - Continuity Rambus Inline Memory Module (Módulo de continuidade de memória em linha Rambus)

É apenas um módulo sem memória



DDR, DDR2, DDR3, DDR4, DDR5

DDR , ou taxa de dados dupla , um chip de memória que processa o dobro de bits de dados por ciclo de clock comparado a um chip de memória RAM padrão. Requer 2,5 volts de energia

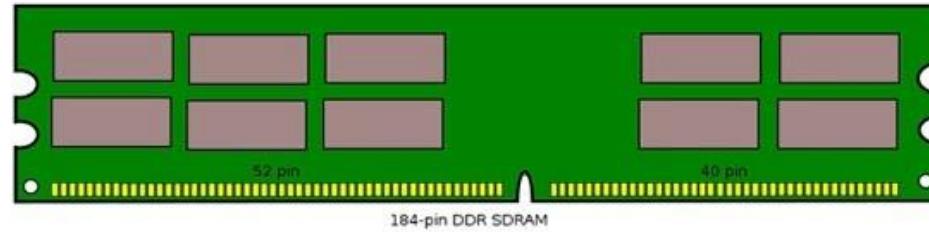
DDR2 é a segunda versão de memória DDR. Requer menos energia - 1,8 volts

DDR3 é a terceira geração de memória double data rate .  
Mais eficiente do que a DDR2. Opera com 1,5 volts de eletricidade.

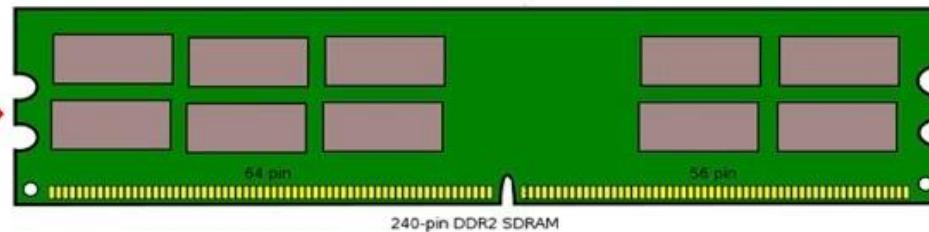
DDR4 – Quarta geração da memória double data rate. Mais eficiente do que a DDR3. – 1,2 volts

DDR, DDR2, DDR3, DDR4

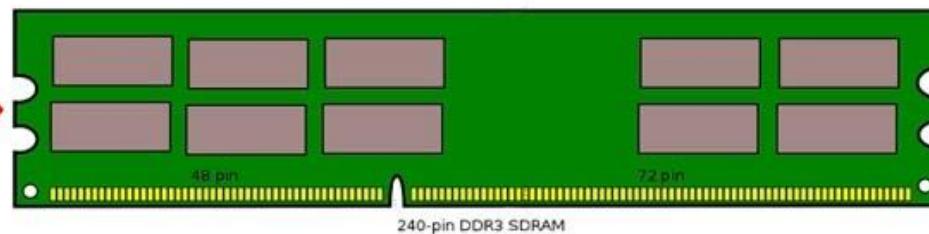
**DDR**



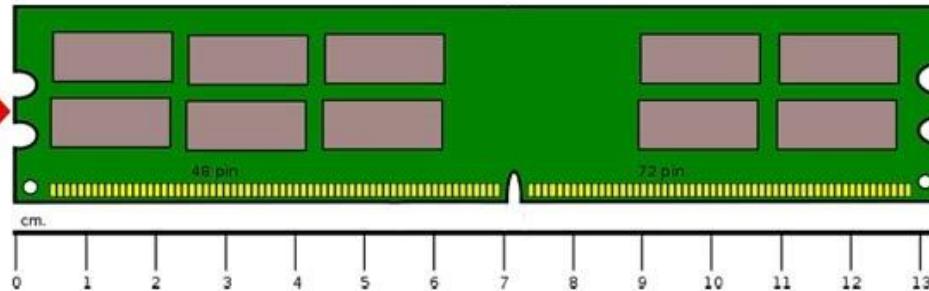
**DDR2**



**DDR3**



**DDR4**



DDR, DDR2, DDR3, DDR4, DDR5

**DDR**



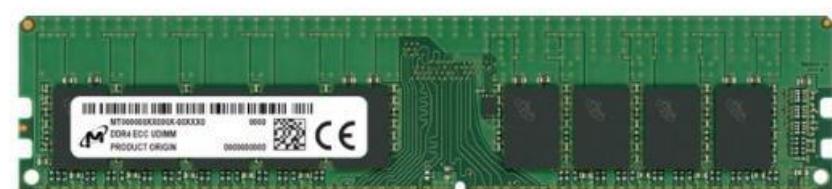
**DDR2**



**DDR3**



**DDR4**



**DDR5**

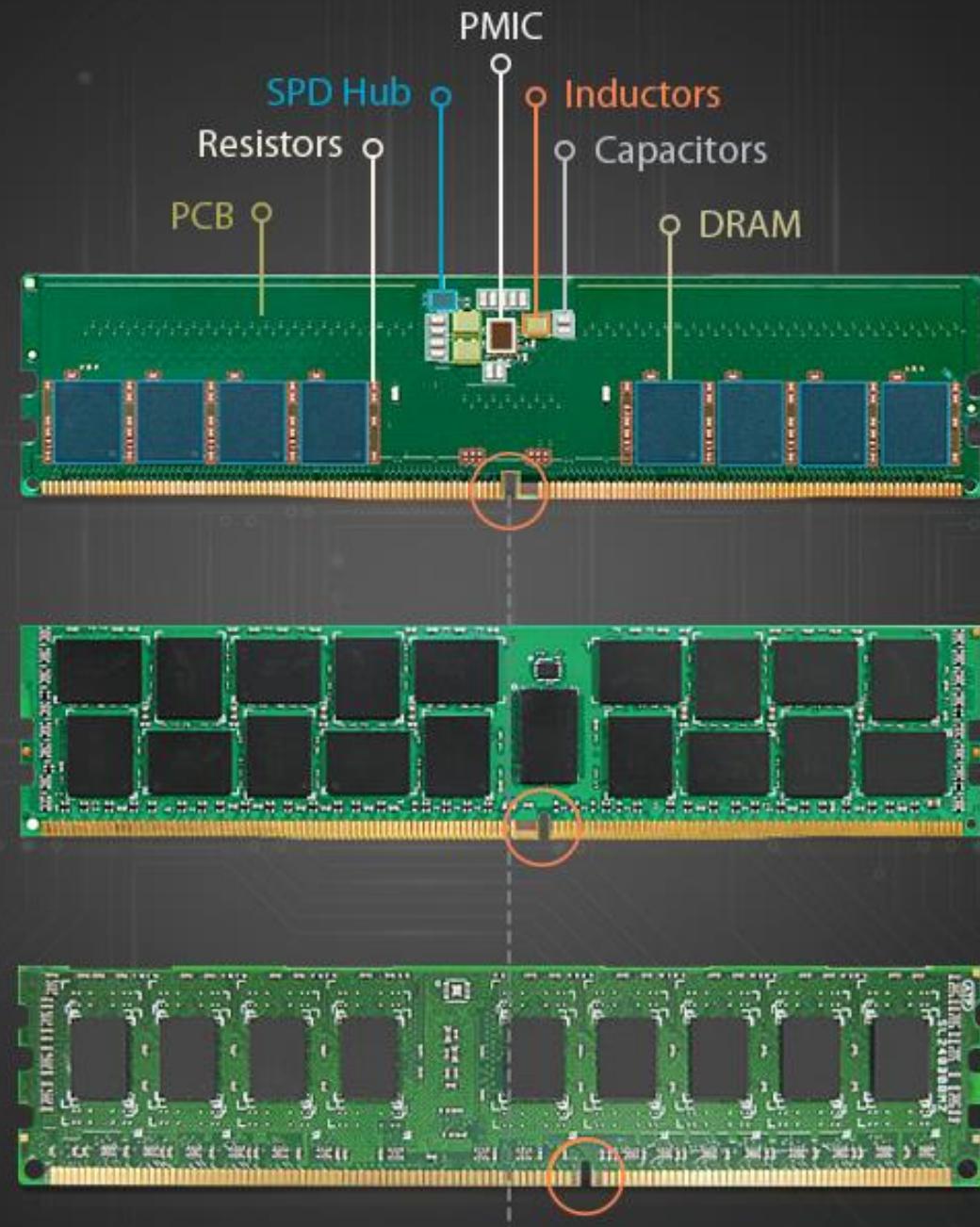


DDR3, DDR4, DDR5

DDR5

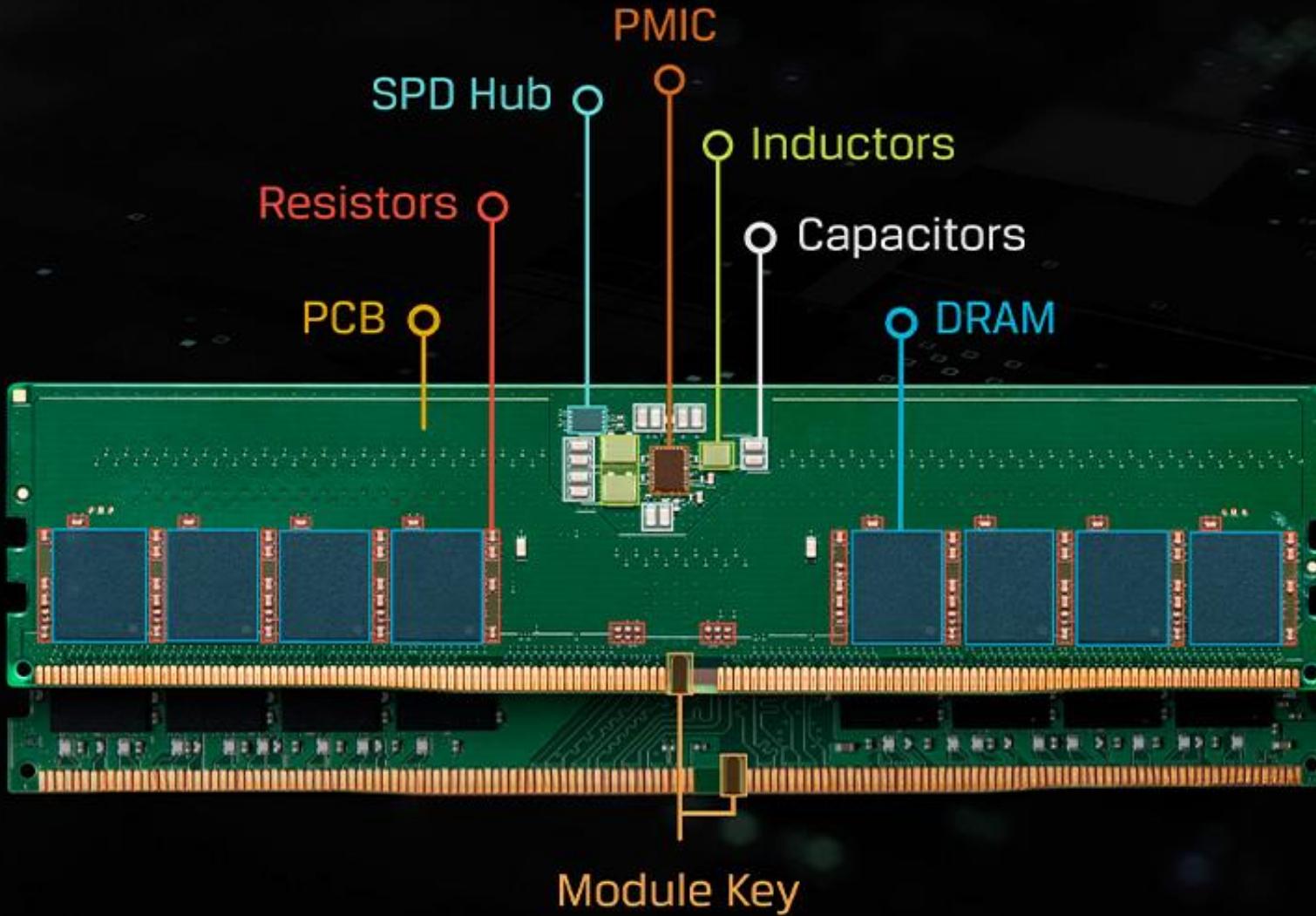
DDR4

DDR3



DDR4, DDR5

*DDR5*  
*DDR4*



# DDR, DDR2, DDR3, DDR4 e DDR5

Memória de acesso aleatório dinâmica	DDR	DDR2	DDR3	DDR4	DDR5
Pré-busca	1 - Bit	2 - Bit	4 - Bit	8 - Bit	Bit por banco
Taxa de dados (Mb/s)	100 - 166	266 - 400	533 - 800	1.066 - 1.600	2.133 - 5.100
Taxa de transferência (GB/s)	0,8 - 1,3	2,1 - 3,2	4,2 - 6,4	8,5 - 14,9	17 - 25,6
Tensão (V)	3,3	2,5 - 2,6	1,8	1,35 - 1,5	1,2
					1,1

# Memórias (RAM e ROM)

ROMs - Read-Only Memory (memória somente de leitura)

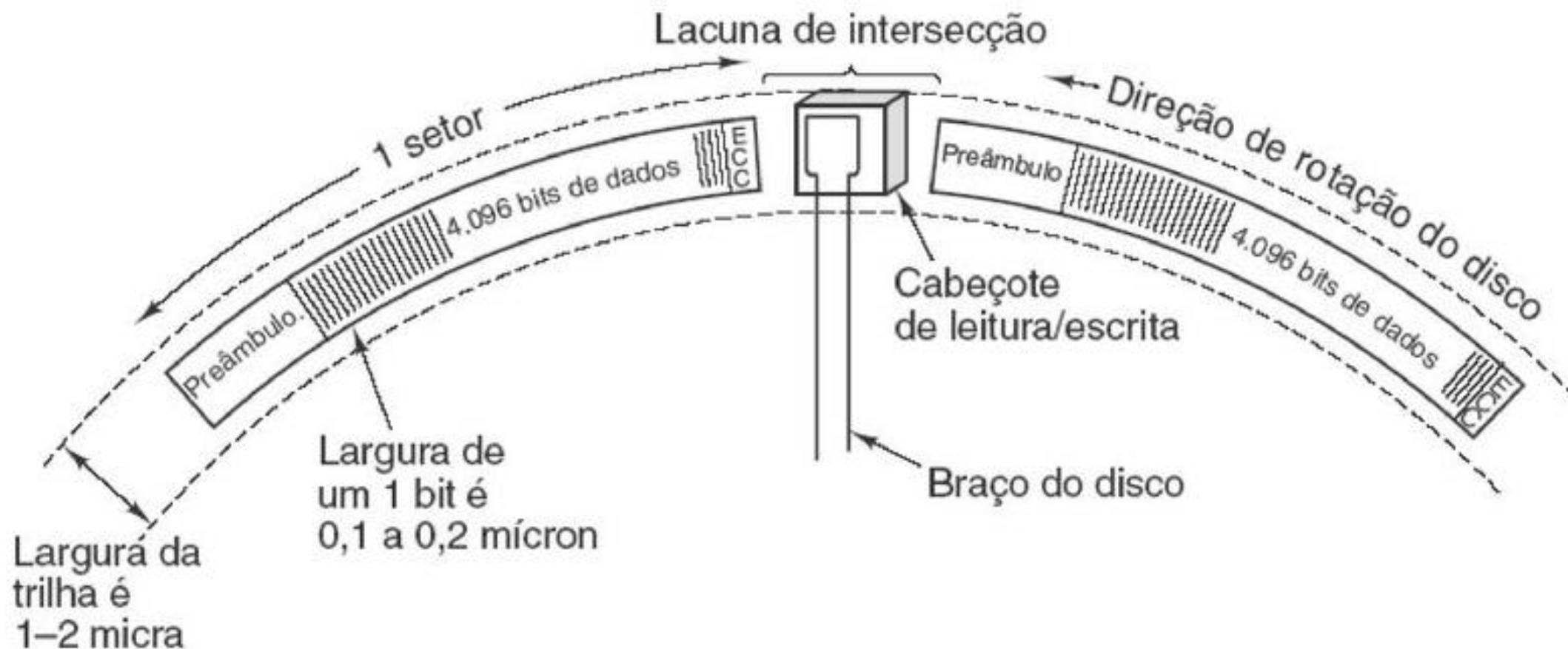
Não volátil – não perde dados / informações

# Memórias (Discos Magnéticos)

- Composto por 1 ou mais pratos de alumínio com um revestimento magnetizável.
- Inicialmente esses pratos tinham 50 cm de diâmetro.
- Atualmente, entre 3 e 12 cm.

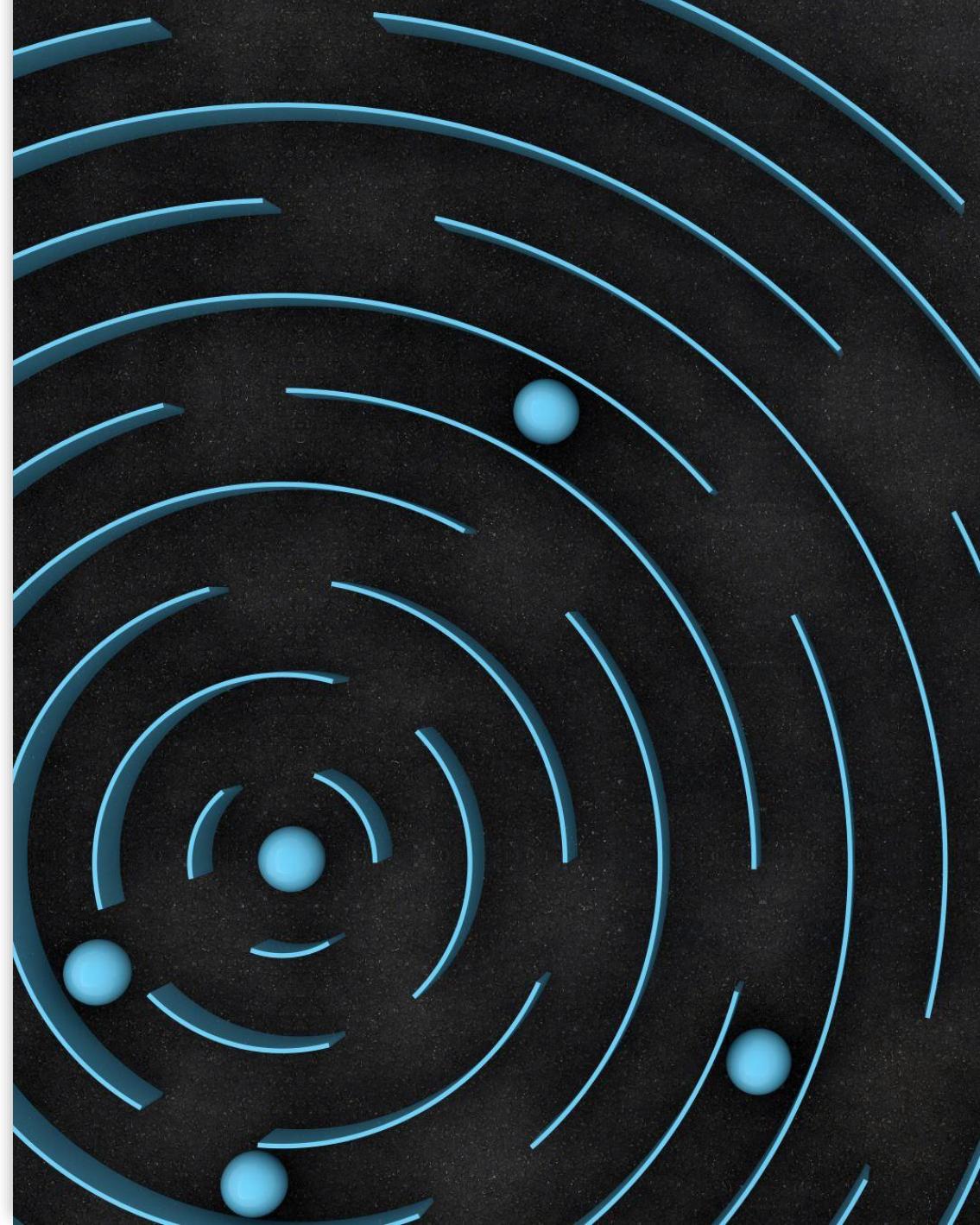


# Memórias (Discos Magnéticos)

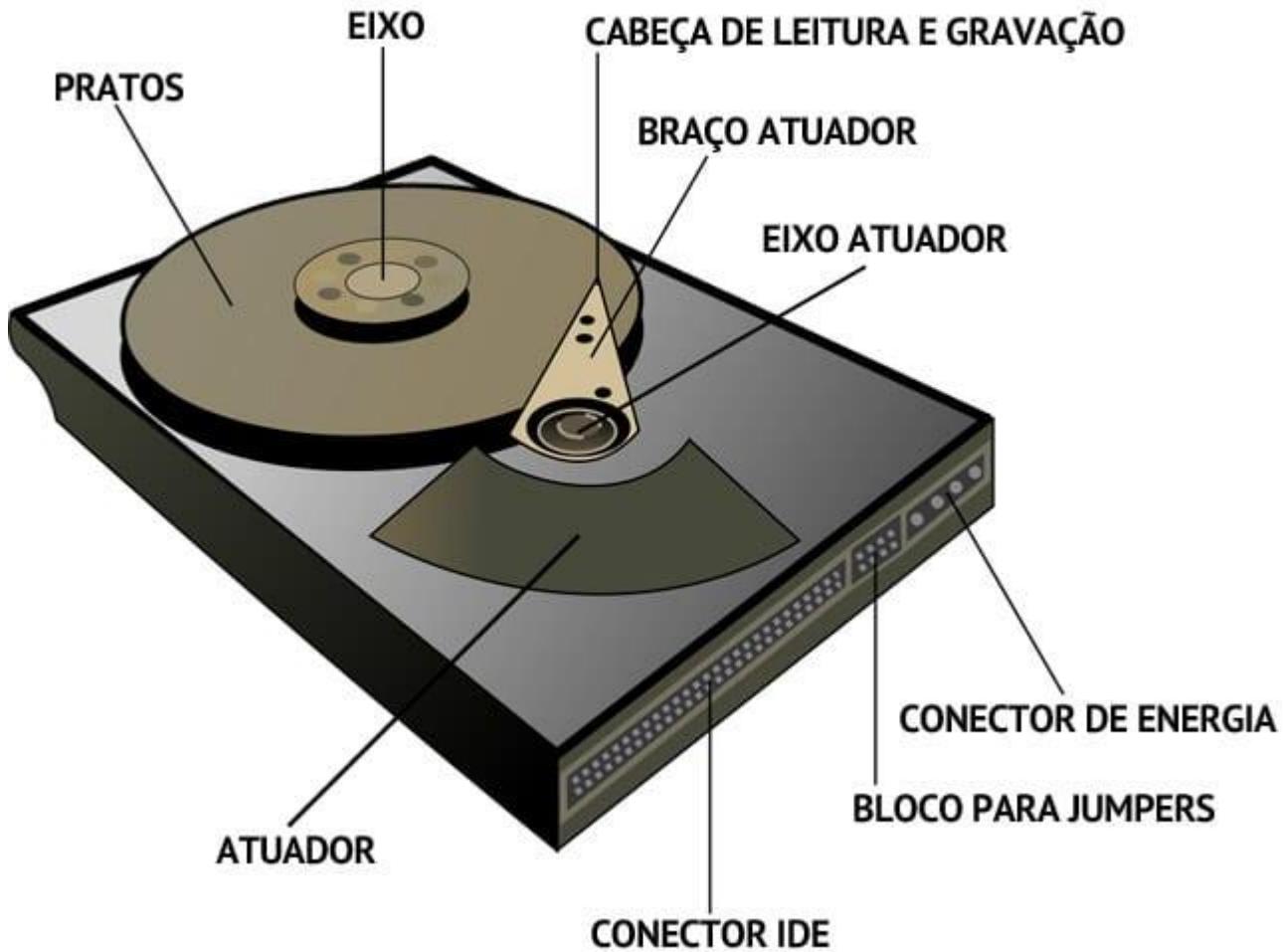
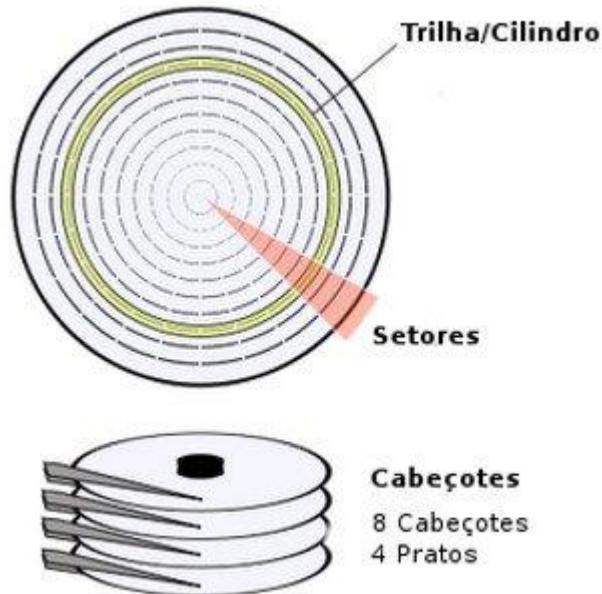


# Memórias (Discos Magnéticos)

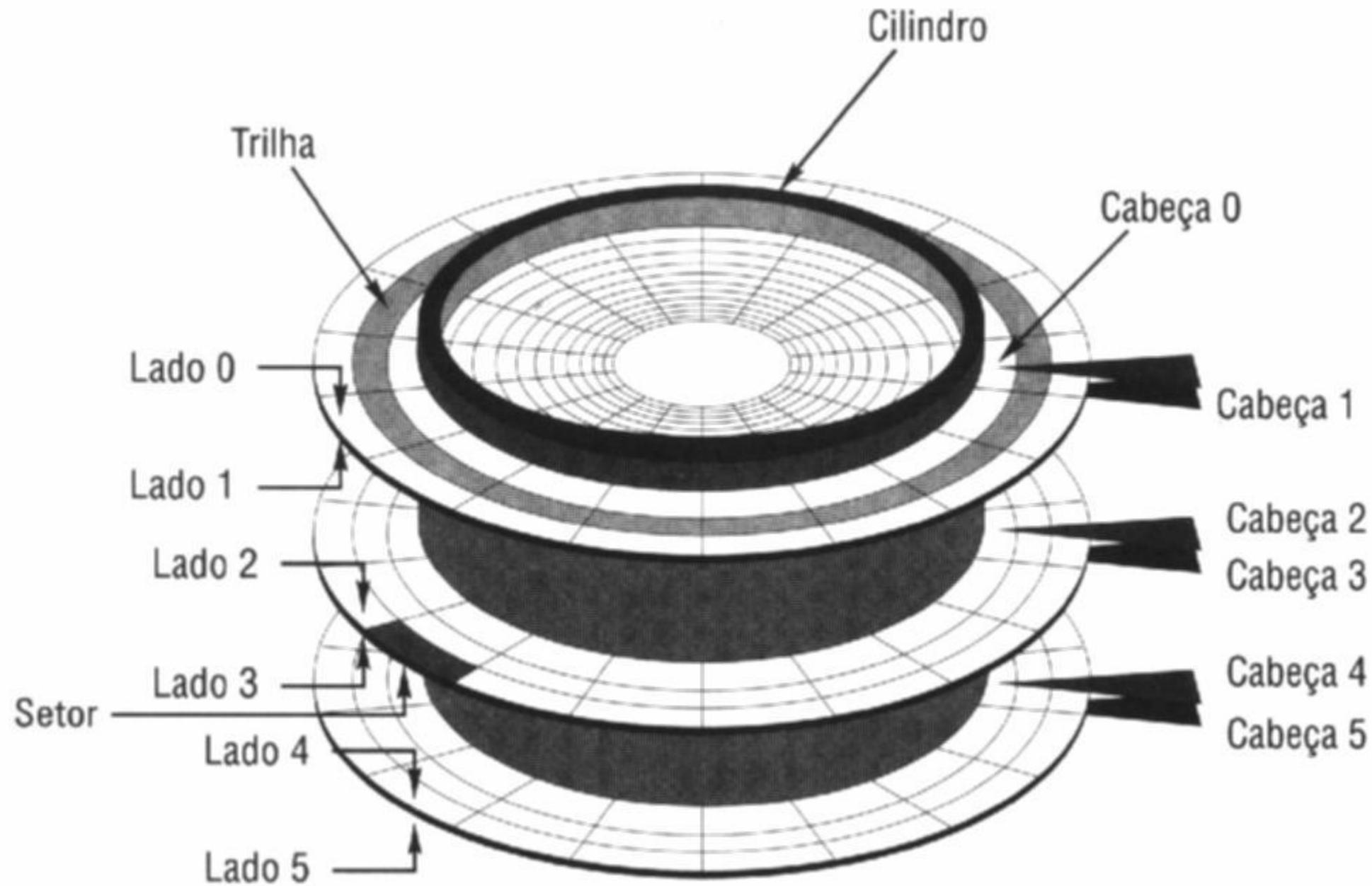
- Trilha
  - Os discos tem aproximadamente 5.000 e 10.000 trilhas por centímetro.
  - Uma trilha não é um sulco na superfície e sim apenas um anel magnetizável.
  - Conjunto de trilhas forma os cilindros.
- Disco
  - Os discos são compostos por vários pratos.
  - Cada superfície do prato tem seu próprio cabeçote de leitura e gravação.



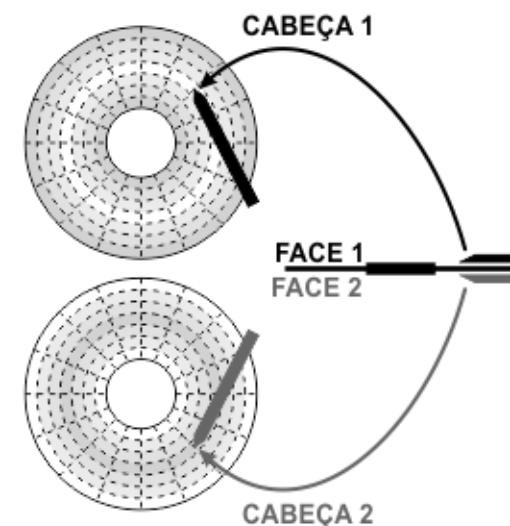
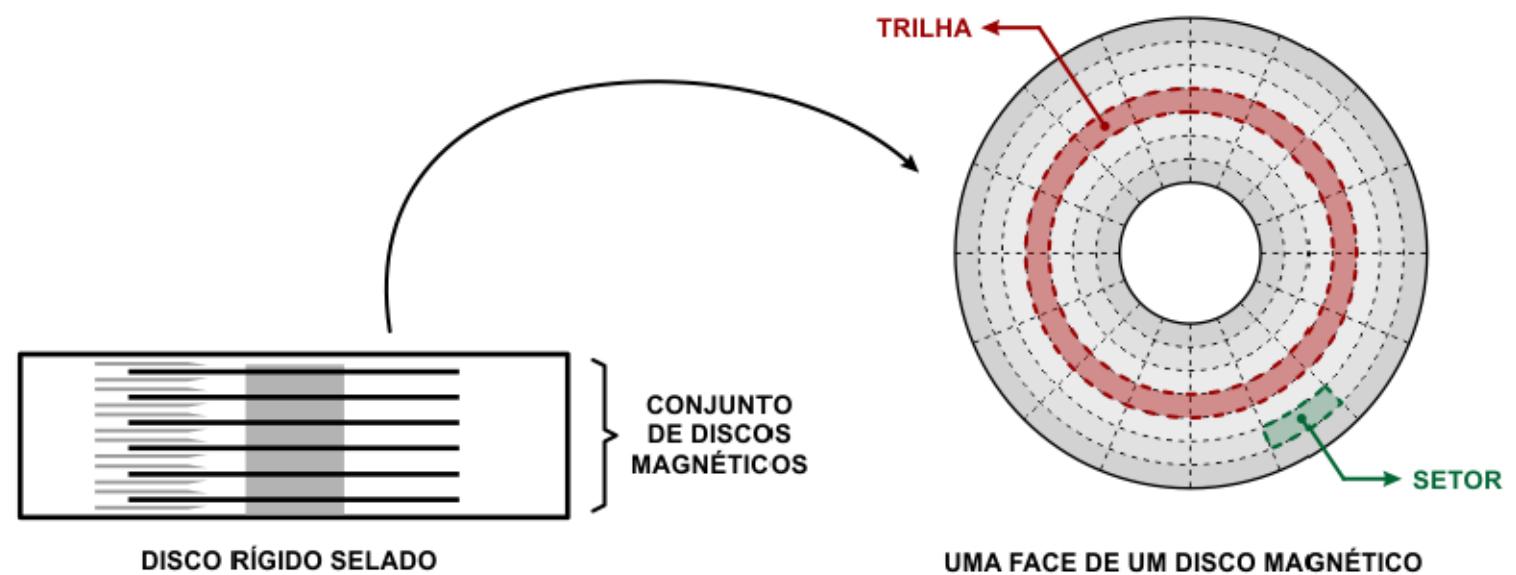
# Memórias (Discos Magnéticos)



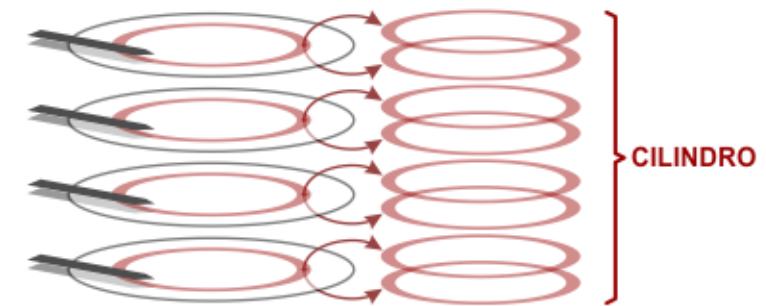
# Memórias (Discos Magnéticos)



# Memórias (Discos Magnéticos)



UMA CABEÇA PARA CADA FACE DE UM DISCO



CILINDRO: CONJUNTO DE TRILHAS SUPERPOSTAS

# Memórias (Discos Magnéticos)

**Setores** – cada trilha é dividida em setores.

Cada setor é precedido de um Preâmbulo.

**Preâmbulo** – permite a sincronização do cabeçote no processo de leitura e escrita.

Cada setor é sucedido de um **ECC**.

ECC – *Error Correcting Code*

Entre os setores há uma lacuna de interseção.

# Memórias (Discos Magnéticos)

## Disco

Os discos possuem uma “CPU” completa.

- Leitura / escrita / formatação
- Movimentação do braço
- Detecção e correção de erros
- Manipulação de buffers
  - Cache de setores com potencial de uso futuro
- Remapeamento de setores ruins

<https://www.youtube.com/watch?v=wtdnatmVdIg>

# Inside a Hard Drive



12 TB

# Inside a Hard Drive

